AN ANALYSIS OF A FREQUENCY COMPRESSIVE RECEIVER

by

Thomas Richard Walsh

B.S., Kansas State University, 1981
M.S., Kansas State University, 1983

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1986

Approved by:

Brian Harms

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

With the ever increasing density of the electromagnetic signal environment, especially in times of hostile activities, there has evolved a need to be able to monitor wide instantaneous RF bandwidths. This monitoring is usually accomplished with surveillance receivers. The surveillance receiver detects the presence of signals and determines the type of modulation and associated parameters. If the signal is agile, the receiver also tracks it. Surveillance receivers are made up of several components. The front end of the surveillance receiver is the intercept receiver. The intercept receiver is primarily responsible for the instantaneous detection of electromagnetic energy in a preselected bandwidth. Once a signal is detected, an estimate of the signal's frequency is made.

Characteristics of a 'good' intercept receiver are that it should maintain a high probability of detection, be able to discriminate between multiple signals that are incident simultaneously, and cover a wide range of frequencies with acceptable resolution.

The main concern of this thesis is to examine one type of intercept receiver, the frequency compressive receiver (FCR). The frequency compressive receiver is also referred to as the microscan receiver.

1.2 Objective and Overview

The objective of this thesis is to analyze the performance of the frequency compressive receiver. The performance will be measured by

the receiver's probability of detection as a function of the signal-to-noise density ratio. Receiver performance will be examined as the parameters that characterize the receiver are varied.

The literature tends to support the notion that post filtering does not improve the performance of the frequency compressive receiver. This thesis examines the performance of the frequency compressive receiver, with and without post filtering, in an attempt to show empirical evidence in support of the above notion.

Chapter Two presents an introduction to the basic principles and operation of the frequency compressive receiver. Current hardware implementations are examined briefly. The remainder of Chapter Two presents the frequency compressive receiver in block diagram form and develops the equivalent lowpass representation of each element. By combining the lowpass representations of each element, a mathematical expression of the receiver output is developed.

Chapter Three is concerned with the development of the probability density function of the output of the frequency compressive receiver. The probability density functions for the cases of noise only and signal plus noise are given. The determination of a threshold voltage, which is directly related to the false alarm rate (FAR) and the noise probability density function, is presented. The chapter concludes with the development of a closed form integral expression describing the probability of detection.

Chapter Four discusses the overall structure and possible applications of a FORTRAN computer program (Appendix B) used to numerically evaluate the probability of detection. The program implements numerical techniques to solve the previously defined expressions. All

the interactively definable parameters are discussed as well as the rationale associated with the default values. Three parameter configurations are examined, subsequent plots of filter outputs, probability density functions, and detection probabilities are presented.

Chapter Five concludes the thesis with a summary of interesting points observed throughout this research.

CHAPTER 2

The Frequency Compressive Receiver

2.1 Introduction

This chapter presents a discussion on the frequency compressive receiver. Basic principles of operation and characteristics of the frequency compressive receiver are examined. Current hardware implementations are also investigated. These include surface acoustic wave (SAW) devices for frequencies in the RF band, and their counterparts for frequencies in the microwave range, namely, magnetostatic wave (MSW) devices.

The remainder of the chapter is devoted to examining a generic block diagram of a frequency compressive receiver and investigating each block individually. The lowpass equivalent representation for each element is developed, as well as a mathematical representation of both its input and output. The chapter concludes with the development of a closed form expression for the overall output of the frequency compressive receiver.

2.2 Operation and Implementation

This section explains the operation and examines various characteristics of the frequency compressive receiver. Current hardware implementations of the dispersive filter necessary in the frequency compressive receiver are also examined.

The frequency compressive receiver is a transform receiver. The receiver computes the magnitude of the Fourier transform of the input signal. The theoretical basis for the frequency compressive receiver is known as the Chirp transform. The Chirp transform is a three step

algorithm for computing a Fourier transform. There are two implementations of the Chirp transform. They are the Multiply-Convolve-Multiply (MCM) and the Convolve-Multiply-Convolve (CMC) algorithms. The input signal is either multiplied (M) or convolved (C) with a 'chirping' function, hence the name. A chirping function is a periodic function whose frequency is a linear function of time. For a more detailed explanation of the Chirp transform the reader is referred to [18] and [2].

Figures (2.2-1a and b) present the block diagrams of the RF model and lowpass model, respectively, of the frequency compressive receiver with post filtering. The frequency compressive receiver consists of a scanning local oscillator (SLO) with output $q(t)$, a dispersive filter with transfer function $H(f)$, and a square-law device. A post filter, $G(f)$, is considered in an attempt to examine its effect on the probability of detection.

The pulse compression inherent in the FCR is a result of the dispersive filter. The dispersive filter delays higher frequencies longer than lower frequencies. Two parameters that characterize the operation of the frequency compressive receiver are the compression bandwidth, β, and the dispersion time, τ. The compression bandwidth is the range of frequencies for which the filter maintains a time delay that is a linear function of frequency. The time delay for the RF model is characterized in Figure (2.2-2a).

Before the signal is incident on the dispersive filter, it is mixed with the output of the SLO. The SLO scans a range of frequencies equal to twice the compression bandwidth in a time interval equal to twice the dispersion time. This results in the

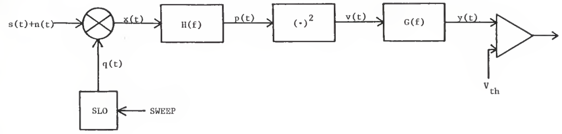


Figure 2.2-1a: RF Model of the Frequency Compressive Receiver

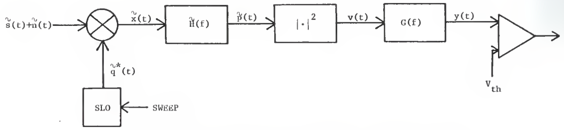


Figure 2.2-1b: Lowpass Model of the Frequency Compressive Receiver

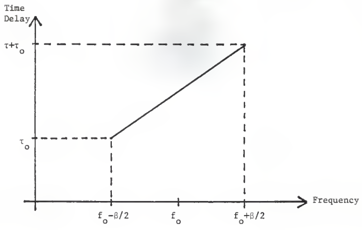


Figure 2.2-2a: Time Delay versus Frequency Characteristics of the Dispersive Filter for the RF Model.

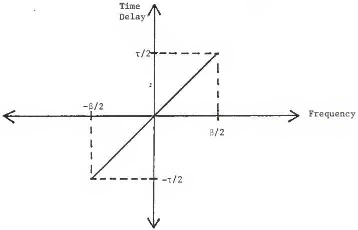


Figure 2.2-2b: Time Delay versus Frequency Characteristics of the Dispersive Filter for the Lowpass Model

higher frequency components of the signal being incident on the dispersive filter prior to lower frequency components. If the SLO and the dispersive filter are matched, then the time delaying of the signal components is such that all the signal energy arrives at the output simultaneously. Thus, the output of the dispersive filter is its impulse response.

The RF operation of the FCR is diagrammed in Figure (2.2-3). The FCR monitors an RF bandwidth, β, with center frequency f_{RF}. It is necessary for the SLO to scan a range equal to twice β so that signals at the upper and lower range of the RF bandwidth experience full compression. The earliest output, in the RF model, occurs at time τ. This corresponds to a signal present at a frequency of $f_{RF} - \beta/2$. Output at time 2τ corresponds to a signal present at a frequency of $f_{RF} + \beta/2$. Output at times between τ and 2τ corresponds to signals with frequency between $f_{RF} - \beta/2$ and $f_{RF} + \beta/2$, respectively. If a signal in the RF band is only present for a time less than the scan time, it is possible that the signal will not experience full compression. If an RF signal is present for a portion of the scan time such that the difference frequencies at the output of the SLO do not pass through the compresaion bandwidth, then no output will be observed.

This thesis analyzes the FCR by examining the lowpass equivalent model (recall Figure (2.2-1b)). The operation of the FCR in terms of its lowpass model is diagrammed in Figure (2.2-4). An interesting phenomenon in the lowpass model is the occurrence of both positive and negative time delays. The lowpass output of the mixer, $\tilde{x}(t)$, consists of both positive and negative frequencies. The non-negative frequen-

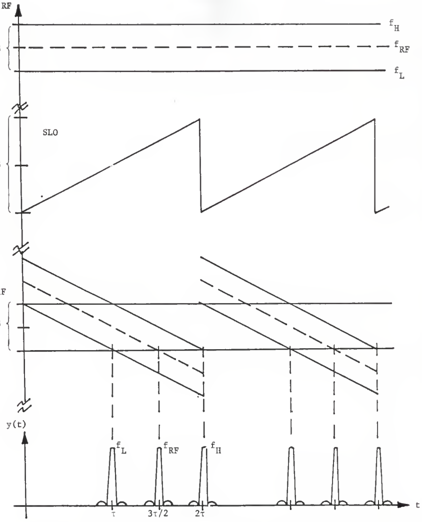


Figure 2.2-3: Frequency-Time Diagram of the RF Model of the Frequency Compressive Receiver

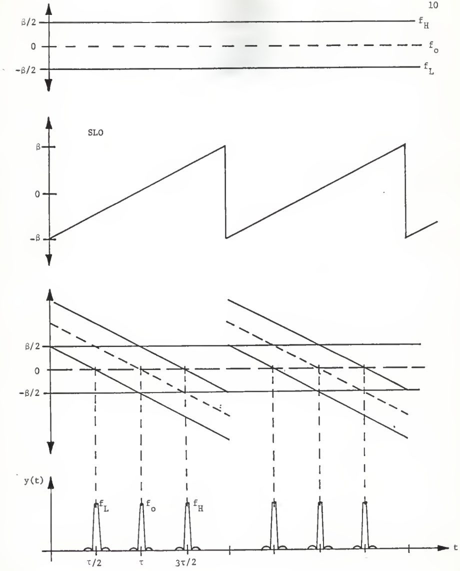


Figure 2.2-4: Frequency-Time Diagram of the Lowpass Model of the Frequency Compressive Receiver

cies experience a time delay that is proportional to their frequency. The negative frequencies result in a negative time delay (see Figure (2.2-2b)). A negative time delay implies that the output of the filter is realized before the frequency component is incident on the filter. This results in observing output prior to time τ. Lowpass representations of bandpass signals, i.e., pulses, with modulation at $-\beta/2$, 0, and $\beta/2$ radians result in filter output at times $\tau/2$, τ, and $3\tau/2$ seconds, respectively.

The resolution of the frequency compressive receiver is an important feature when considering its operation in a dense signal environment. The resolution is a function of the dispersive filter's impulse response, which is apparent from Figure (2.2-3 and 4), as well as the duration of time that the signal is incident on the receiver.

If it is assumed that the dispersive filter has a rectangular magnitude response with bandwidth, B, and compression bandwidth, β, then it can be shown [2] that the signals must be separated by

$$\frac{1}{B}\ (\beta/\tau)\ \text{Hz} \tag{2.2.1}$$

in order to be resolved. If the 3-dB bandwidth of the dispersive filter is assumed to be the same as the compression bandwidth, then resolution (2.2.1) is reduced to

$$\text{Resolution} = 1/\tau\ \text{Hz}. \tag{2.2.2}$$

For a signal that is present for the entire scan time, 2τ, the gating effect of the SLO will make it appear as a τ second pulse. The output of the filter, the Fourier transform of a τ second pulse, will be a sinc function with zeroes at integer multiples of $1/\tau$. Thus, resolution is again given as

$$\text{Resolution} = 1/\tau\ \text{Hz}. \tag{2.2.3}$$

The FCR has a Figure of merit associated with it. It is referred to as the filter's compression factor, F. The compression factor is defined as the ratio of the duration of an uncompressed pulse, τ seconds, to the duration of a compressed pulse, $1/\beta$ to $2/\beta$ seconds assuming that $\beta = B$, which is given as

$$F = \frac{\tau}{1/\beta} = \tau\beta \ . \qquad (2.2.4)$$

Because of the form of equation (2.2.4), the compression factor is commonly refered to as the filter's time bandwidth product.

At this point, it is appropriate to briefly introduce current hardware implementations for obtaining the dispersive characteristics necessary in the frequency compressive receiver. Two resources for implementing pulse compression are surface acoustic wave (SAW) devices [8], [11], [12], [13], [16], [17], and [18] and magnetostatic wave (MSW) devices [3], [4], and [5].

SAW devices, which are used for frequencies in the UHF/VHF band, can be realized by two different approaches. In one approach, SAW devices are based on interdigital electrode transducers (IDT) [18]. Dispersion is obtained by varying the electrode spacing across the IDT, where the electrodes are interleaved and deposited on a piezoelectric substrate. By varying the electrode overlap, it is possible to vary the amplitude versus time and frequency versus time relationships independently, thus providing a wide variety of chirping functions. Figure (2.2-5a) shows an in-line interdigital transducer. Figure (2.2-5b) shows an inclined IDT where the incline helps to reduce spurious signal action.

SAW devices can also be realized by the reflective array compressor (RAC) design, [18]. The RAC design separates the input and

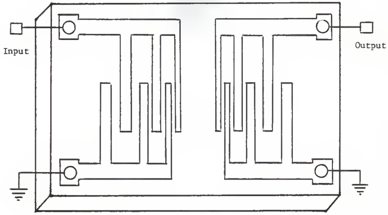


Figure 2.2-5a: In-Line Interdigital Transducer

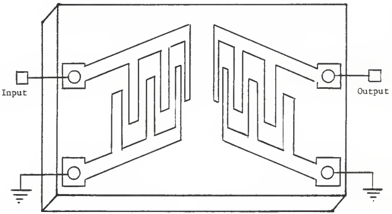


Figure 2.2-5B: Inclined Interdigital Transducer

output transducers so as to allow independent optimization of both of these functions. Figure (2.2-6) shows an etched groove reflective array device. The grooved arraya are etched into a piezoelectric substrate. The filter's amplitude response is determined by the variations in the groove depth. The filter's frequency responae is a function of the groove separation distance. A metalized phase plate allows for the compensation of phase errors after fabrication. The RAC devices are capable of achieving higher time bandwidth products than devices based on IDT devicea. The RAC devices are currently obtaining time bandwidth products from 10 to 50,000, whereaa the IDT design are only achieving time bandwidth products of 4 to 2000, [18]. However, the IDT designa tend to be more stable.

At microwave frequencies, a new technology based on magnetostatic wave (MSW) propagation in magnetically biased epitaxial films of yttrium iron garnet (YIG) has evolved for the implementation of linear diapersive delay lines. Magnetostatic waves are slow, dispersive, magnetically dominated electromagnetic waves which propagate in magnetically biased ferrite materials. Magnetostatic wavea, which can propagate in a ferromagnetic film, exhibit nonlinear disperaion. It is possible to obtain linear dispersion by modifying the geometry of the boundary conditions for the magnetostatic wave.

For example [4], the position of the ground plane relative to the plane of the magnetic film significantly changes the group delay veraus frequency characteristics. A structure uaing one YIG film and a ground plane is shown to yield the theoretically expected linear delay of 50 to 230 ns/cm over a 16 GHz bandwidth at the X-band [4].

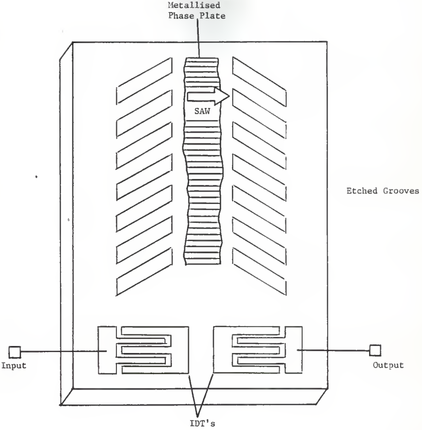


Figure 2.2-6: Etched Groove Reflective Array Device

Also, Bangianni [19] showed a way of obtsining a linear delay of frequencies using a sandwich structure of two YIG films.

2.3 Signals Incident on the Receiver

The signal incident on the receiver is the sum of an RF bandpass signal, s(t), and a stationary bandpass noise process, n(t). It is assumed that the noise process is white over the appropriate range of frequencies. It is difficult to analyze data at such high frequencies because the number of sample points required to accurately represent the signal becomes to large for computer analysis. For our purposes, the information of interest is in the envelope of the signal, therefore, the lowpass representation of these signals will be used in the model.

The complex envelopes [1] of the lowpass equivalent model are related to the bandpass signals by

$$s(t) = \mathrm{Re}\left\{\tilde{s}(t)e^{j\omega_c t}\right\} \tag{2.3.1}$$

$$n(t) = \mathrm{Re}\left\{\tilde{n}(t)e^{j\omega_c t}\right\} \tag{2.3.2}$$

where

$s(t)$, $n(t)$ are the bandpass signals,

$\tilde{s}(t)$, $\tilde{n}(t)$ are the complex envelopes,

and

ω_c is the center frequency of the bandpass signals.

The complex envelopes, $\tilde{s}(t)$ and $\tilde{n}(t)$, are both represented by finite series expansions. The complex envelope of the bandpass signal, $\tilde{s}(t)$, is represented by the Fourier series of its periodic extension,

$$\tilde{s}(t) = \sum_{n=-N}^{N} s_n e^{j2\pi f_n t} \tag{2.3.3}$$

where

$\tilde{s}(t)$ is the complex envelope of $s(t)$,

s_n are the DFT coefficients of $\tilde{s}(t)$,

and

f_n are harmonically related frequencies.

The complex envelope of the stationary bandpass noise process, $\tilde{n}(t)$, is also repreaented by a finite series expansion,

$$\tilde{n}(t) = \sum_{k=-K}^{K} n_k \, e^{j2\pi f_k t} \qquad (2.3.4)$$

where

$\tilde{n}(t)$ is the complex envelope of $n(t)$,

$n_k = n_{k_r} + jn_{k_i}$,

$n_{k_r} \sim N(0, R_{k/2})$,

$n_{k_i} \sim N(0, R_{k/2})$,

$n_k \sim N(0, R_k)$,

and

f_k are frequencies not harmonically related.

It can be shown that the autocorrelation function of $\tilde{n}(t)$ is as follows,

$$R_{\tilde{n}}(\tau) = \sum_{k=-K}^{K} R_k \, e^{j2\pi f_k t} \qquad (2.3.5)$$

The Gaussian quadrature rule may also be used to represent the autocorrelation function as the inverse transform of the power spectrum. Equating these two representations, the autocorrelation function can be expressed in terma of the Gausaian quadrature rule [2] coefficienta, $[\gamma_k, \nu_k]$, aa

$$R_{\tilde{n}}(\tau) = \sum_{k=-K}^{K} 2B_n N_o \gamma_k \, e^{j2\pi B_n \nu_k t} \qquad (2.3.6)$$

where

B_n is the equivalent lowpass noise model bandwidth,

N_o is the noise density,

and

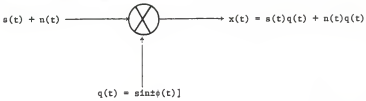
$[\gamma_k, \nu_k]_{k=-K}^{K}$ correspond to a GQR on [-1,1] with unit weighting function.

In conclusion, the lowpass representation of the signal incident on the frequency compressive receiver is denoted as

$$\tilde{s}(t) + \tilde{n}(t). \tag{2.3.7}$$

2.4 Scanning Local Oscillator

The scanning local oscillator (SLO) generates a chirping function. A chirping function is a periodic function whose frequency is a linear function of time, or equivalently, the function has quadratic phase characteristics. The mixing operation results in a down-chirp signal being present at the output. A down-chirp signal is a periodic signal whose frequency is a linearly decreasing function of time. The RF representation of this situation is illustrated in Figure (2.4-1).



where	$s(t),n(t)$	are the incident bandpass signals.
	$q(t)$	is an up-chirp local oscillator,
and		
	$x(t)$	is the down-chirp output.

FIGURE 2.4-1: RF Model of a Scanning Local Oscillator

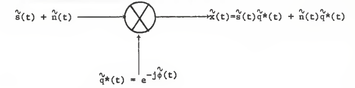
It can be shown that the lowpass representation of the RF SLO, q(t), is given by

$$\tilde{q}(t) = e^{j\tilde{\phi}(t)} \tag{2.4.1}$$

where

$\tilde{\phi}(t)$ is quadratic in t.

For the frequency compressive receiver to operate correctly, the SLO must be matched to the dispersive filter, i.e., the complex conjugate of $\tilde{q}(t)$ is used. The lowpass equivalent representation of Figure (2.4-1) is shown in Figure (2.4-2).



where

$\tilde{s}(t)$, $\tilde{n}(t)$ are lowpass incident signals,

$\tilde{q}^*(t)$ is the conjugate of the lowpass SLO,

and

$\tilde{x}(t)$ is the lowpass down chirp signal.

FIGURE 2.4-2: Lowpass Model of the Scanning Local Oscillator

The instantaneous frequency of the scanning local oscillator is given as

$$\frac{d\tilde{\phi}(t)}{dt} = \frac{d}{dt}\left[at^2 + bt + c\right] = 2at + b. \tag{2.4.2}$$

The scanning local oscillator must sweep a range of frequencies equal to twice the compression bandwidth in a time period T, where T is the period of the lowpass signal incident on the receiver. In the lowpass

model, Figure (2.2-4), it follows that equation (2.4.2) must satisfy the following

$$\left.\frac{d\tilde{\phi}(t)}{dt}\right|_{t=0} = 2at + b = -\beta_{\omega} \ , \tag{2.4.3a}$$

$$\left.\frac{d\tilde{\phi}(t)}{dt}\right|_{t=\tau} = 2at + b = 0 \ , \tag{2.4.3b}$$

and

$$\left.\frac{d\tilde{\phi}(t)}{dt}\right|_{t=T} = 2at + b = \beta_{\omega} \ . \tag{2.4.3c}$$

Solving equations (2.4.3a-c) yields the following

$$a = \frac{2\pi \ \beta_{Hz}}{T} = \frac{\beta_{\omega}}{T} \ , \tag{2.4.4a}$$

and

$$b = -2\pi \ \beta_{Hz} = -\beta_{\omega} \ . \tag{2.4.4b}$$

Substituting equations (2.4.4a-b) back into (2.4.2) yields the following

$$\frac{d\tilde{\phi}(t)}{dt} = 2 \frac{\beta_{\omega}}{T} t - \beta_{\omega} \ . \tag{2.4.5}$$

The phaae, $\tilde{\phi}(t)$, of the scanning local oscillator is obtained by integrating equation (2.4.5) with respect to time, i.e.,

$$\tilde{\phi}(t) = \int \left(2 \frac{\beta_{\omega}}{T} t - \beta_{\omega}\right) dt$$

$$= \frac{\beta_{\omega}}{T} t^2 - \beta_{\omega} t + c \ . \tag{2.4.6}$$

Rewrite equation (2.4.6) in terms of scanning rates; the phase is now

$$\tilde{\phi}(t) = \frac{1}{2} s_{\omega} t^2 - \tau s_{\omega} t + c \tag{2.4.7}$$

where

s_{ω} is the radian scanning frequency,

τ is the filter dispersion time,

and

c is a constant of integration.

The lowpass output of the scanning local oscillator mixed with the incident signal is

$$\tilde{x}(t) = \tilde{s}(t)\tilde{q}^*(t) + \tilde{n}(t)\tilde{q}^*(t)$$

$$= \tilde{s}(t)\ e^{-j\phi(t)} + \tilde{n}(t)\ e^{-j\phi(t)}$$

$$= \sum_{n=-N}^{N} s_n\ e^{-j\phi(t)}\ e^{j2\pi f_n t} + \sum_{k=-K}^{K} n_k\ e^{-j\phi(t)}\ e^{j2\pi f_k t}. \quad (2.4.8)$$

Since the incident signal and the scanning local oscillator are both periodic with the same period, the signal component can still be represented by the Fourier series of its periodic extension with its DFT coefficients scaled appropriately.

The mixer introduces a phase shift in the noise process that is incident on the receiver, (2.3.4). This is equivalent to a shift in frequency of the power spectrum of the incident noise process. The noise process at the input of the receiver is assumed to be white and band-limited. The power spectrum is given as follows

$$S_{\tilde{n}}(f) = \begin{cases} 2N_o & |f| < B_n \\ 0 & |f| > B_n\ . \end{cases} \quad (2.4.9)$$

Since the scanning local oscillator is periodic, the power spectrum of the noise at the output of the scanning local oscillator is also periodic. The power spectrum at times t = 0, T/2, and T, where T is the period of the SLO, are given as

$$S_{\tilde{n}}(f) = \begin{cases} 2N_o & -B_n - \beta \leq f \leq B_n - \beta \\ 0 & \text{otherwise,} \end{cases} \quad (2.4.10a)$$

$$S_{\tilde{n}}(f) = \begin{cases} 2N_o & -B_n \leq f \leq B_n \\ 0 & \text{otherwise,} \end{cases} \quad (2.4.10b)$$

and

$$S_{\tilde{n}}(f) = \begin{cases} 2N_o & -B_n + \beta < fn < B_n + \beta \\ 0 & \text{othsrwise.} \end{cases} \qquad (2.4.10c)$$

The value of the lowpass noise model bandwidth, B_n, is dependent on β, and must be choosen such that the noise power spectrum covers a band of frequencies just wider than the noise bandwidth of the dispersive filter. If the equivalent noise model bandwidth, B_n, is chosen appropriately, then the mixing operation still results in a band limited white noise process at the output of the scanning local oscillator. Therefore, the mixer will not be taken into account in terms of the noise component and equation (2.4.8) may be rewritten as

$$\tilde{x}(t) = \sum_{n=-N}^{N} s_n e^{j2\pi f_n t} + \sum_{k=-K}^{K} n_k e^{j2\pi f_k t} \qquad (2.4.11)$$

where

s_n is the n^{th} DFT coefficient of $\tilde{s}(t)e^{-j\tilde{\phi}(t)}$,

f_n is the n^{th} harmonic frequency,

$n_k = n_{k_r} + jn_{k_i}$,

$n_{k_r} \sim N(0, R_{k/2})$,

$n_{k_i} \sim N(0, R_{k/2})$,

$n_k \sim N(0, R_k)$,

and

f_k are frequencies determined by the GQR.

The appropriate value of B_n is considered again in Chapter Four.

2.5 Dispersive Filter

This section develops the lowpass representation of the dispersive filter and mathematically describes its output. The lowpass dispersive filter, $\tilde{H}(\omega)$, is defined as

$$\tilde{H}(\omega) = H(\omega - \omega_c)_{LP}$$

$$= \tilde{Y}(\omega)\, e^{j\tilde{\phi}(\omega)} \tag{2.5.1}$$

where

$H(\cdot)$ is the RF dispersive filter transfer function,

$\tilde{H}(\cdot)$ is the lowpass dispersive filter transfer function,

$\tilde{Y}(\cdot)$ is the lowpass magnitude response,

and

$\tilde{\phi}(\omega)$ is the lowpass phase response.

The dispersive filter has a Gaussian magnitude response and quadratic phase. The Gaussian magnitude response is used in an effort to obtain an impulse response with reduced sidelobes, thus resulting in increased resolution for detecting signals close in frequency.

The magnitude response, $\tilde{Y}(\omega)$, is defined as

$$\tilde{Y}(\omega) = A\, e^{-a\omega^2} \tag{2.5.2}$$

where

A is the filter gain at DC,

a is determined by the cutoff frequency,

and

ω are frequencies such that $-\beta/2 < \omega < \beta/2$.

The filter is assumed to have unity gain at DC, i.e., A=1. The exponential constant, a, is obtained as follows,

$$\tilde{Y}\left(\frac{\omega}{\omega_c}\right)\Big|_{\omega=\omega_c} = e^{-a\left(\frac{\omega}{\omega_c}\right)^2}$$

$$= e^{-a}$$

$$= .7071.$$

Solving for the exponential constant yields the following for the magnitude response,

$$\tilde{Y}(\omega) = e^{-.34657\omega^2} . \tag{2.5.3}$$

The dispersive filter has quadratic phase characteristics so as to insure a time delay, both positive and negative, that is a linear function of frequency. The instantaneous time delay is given as

$$\text{Time Delay} = \frac{-d\tilde{\phi}(\omega)}{d\omega} = \frac{-d}{d\omega}\,[a\omega^2 + b\omega + c] \tag{2.5.4}$$
$$= -(2a\omega + b)$$

where

$\tilde{\phi}(\omega)$ is the phase response of $\tilde{H}(\omega)$.

To obtain the operation of the frequency compressive receiver as depicted by Figure (2.4-4), equation (2.5.4) must satisfy the following conditions,

$$\left.\frac{-d\tilde{\phi}(\omega)}{d\omega}\right|_{\omega=\beta_\omega/2} = -(2a\omega + b) = \tau/2 , \tag{2.5.5a}$$

$$\left.\frac{-d\tilde{\phi}(\omega)}{d\omega}\right|_{\omega=0} = -(2a\omega + b) = 0, \tag{2.5.5b}$$

and

$$\left.\frac{-d\tilde{\phi}(\omega)}{d\omega}\right|_{\omega=-\beta_\omega/2} = -(2a\omega + b) = -\tau/2 . \tag{2.5.5c}$$

Solving equations (2.5.5a-c) yields the coefficients

$$a = \frac{-\tau}{2\beta_\omega} , \tag{2.5.6a}$$

and

$$b = 0. \tag{2.5.6b}$$

Substituting equations (2.5.6a,b) back into (2.5.4) yields the following expression for the time delay of the lowpass dispersive filter, given by

$$\frac{-d\tilde{\phi}(\omega)}{d\omega} = \frac{\tau}{\beta_\omega}\,\omega . \tag{2.5.7}$$

A graphical representation of equation (2.5.7) was shown previously in Figure (2.2-2b).

The phase of the dispersive filter, $\tilde{\phi}(\omega)$ is obtained by integrating equation (2.5.7) with respect to frequency, i.e.,

$$\tilde{\phi}(\omega) = \int \frac{\tau}{\beta_\omega} \omega \, d\omega$$

$$= \frac{1}{2} \frac{\tau}{\beta_\omega} \omega^2 + c \tag{2.5.8}$$

It should be noted that equation (2.5.7) is valid for positive and negative values of ω. This implies the presence of both positive and negative time delays, which was discussed previously with respect to Figure (2.2-4).

The output of the dispersive filter is computed by modifying the DFT coefficients. The output is given as

$$\tilde{p}(t) = \sum_{n=-N}^{N} s_n \tilde{H}(f_n) e^{j2\pi f_n t} + \sum_{k=-K}^{K} n_k \tilde{H}(f_k)\, e^{j2\pi f_k t} \tag{2.5.9}$$

$$= \tilde{P}_S + \tilde{P}_N .$$

where $\tilde{P}_S$ is the lowpass signal component,

and $\tilde{P}_N$ is the lowpass noise component.

2.6 Square-Law Device

The output of the dispersive filter is the input to the square-law device. The lowpass equivalent of a square-law device is just the magnitude squared of the input signal. The output of the square-law device is given as

$$
\begin{aligned}
v(t) &= |\tilde{p}(t)|^2 \\
&= (\tilde{P}_S + \tilde{P}_N)(\tilde{P}_S^* + \tilde{P}_N^*) \\
&= \sum_{n=-N}^{N} \sum_{m=-N}^{N} s_n s_m^* \tilde{H}(f_n)\tilde{H}(f_m^*)\, e^{j2\pi(f_n-f_m)t} \\
&\quad + \sum_{k=-K}^{K} \sum_{\ell=-K}^{K} n_k n_\ell^* \tilde{H}(f_k)\tilde{H}(f_\ell^*) e^{j2\pi(f_k-f_\ell)t} \\
&\quad + \sum_{n=-N}^{N} \sum_{k=-K}^{K} s_n n_k^* \tilde{H}(f_n)\tilde{H}(f_k^*) e^{j2\pi(f_n-f_k)t} \\
&\quad + \text{conjugate of previous term.} \qquad (2.6.1)
\end{aligned}
$$

2.7 Post Filter and Output

The last element considered in the frequency compressive receiver model is the lowpass post filter. The output of this filter is obtained by modifying Fourier series coefficients at the frequency components present in the signal after passing thru the square-law device. The expression for the receiver output is

$$
\begin{aligned}
y(t) &= \sum_{n=-N}^{N} \sum_{m=-N}^{N} s_n s_m^* \tilde{H}(f_n)\tilde{H}(f_m^*) G(f_n-f_m) e^{j2\pi(f_n-f_m)t} \\
&\quad + \sum_{k=-K}^{K} \sum_{\ell=-K}^{K} n_k n_\ell^* \tilde{H}(f_k)\tilde{H}(f_\ell^*) G(f_k-f_\ell) e^{j2\pi(f_k-f_\ell)t} \\
&\quad + \sum_{n=-N}^{N} \sum_{k=-K}^{K} s_n n_k \tilde{H}(f_n)\tilde{H}(f_k^*) G(f_n-f_k) e^{j2\pi(f_n-f_k)t} \\
&\quad + \text{conjugate of previous term.} \qquad (2.7.1)
\end{aligned}
$$

With the use of matrices, equation (2.7.1) can be written as

$$y(t) = \underline{S}^H \underline{TS} = \underline{N}^H \underline{PN} + \underline{N}^H \underline{QS} + \underline{S}^H \underline{Q}^H \underline{N} \qquad (2.7.2)$$

where

$$\underline{S} = [s_{-N}, \ldots, s_0, \ldots, s_N]^T , \qquad (2.7.2a)$$

$$\underline{N} = [n_{-K}, \ldots, n_0, \ldots, n_K]^T, \qquad (2.7.2b)$$

$$\underline{T} = [\tau_{m,n} = \tilde{H}(f_n)\tilde{H}(f_m^*)G(f_n-f_m)e^{j2\pi(f_n-f_m)t}], \tag{2.7.2c}$$

$$\underline{P} = [\rho_{k,\ell} = \tilde{H}(f_\ell)\tilde{H}(f_k^*)G(f_\ell-f_k)e^{j2\pi(f_\ell-f_k)t}], \tag{2.7.2d}$$

$$\underline{Q} = [q_{k,n} = \tilde{H}(f_n)\tilde{H}(f_k^*)G(f_n-f_k)e^{j2\pi(f_n-f_k)t}], \tag{2.7.2e}$$

and

H denotes hermitian transpose.

CHAPTER 3

THE PROBABILITY OF DETECTION

3.1 Introduction

This chapter presents a mathematical development of the probability density function for the output of the FCR receiver. The probability density functions for the cases of noise only and signal plus noise are considered. The procedure of solving for a particular threshold value, given an arbitrary false alarm rate is presented. Lastly, a closed form integral expression for the probability of detection, as a function of the previously defined threshold, is presented.

3.2 Characteristic Function of the Output

Recall from chapter two that the output of the receiver, y(t), is expressed as

$$y(t) = \underline{S}^H \underline{TS} + \underline{N}^H \underline{PN} + \underline{N}^H \underline{QS} + \underline{S}^H \underline{Q}^H \underline{N}. \qquad (3.2.1)$$

The probability density function of y(t) is found by taking the Fourier transform of the characteristic function of equation (3.2.1). The characteristic function of the output, y(t), is defined as, $M_y(\nu)$, where

$$M_y(\nu) = E[e^{j\nu y}] = \int_0^\infty e^{j\nu y} f_Y(y)\, dy \; . \qquad (3.2.2)$$

It can be observed that the exponent in (3.2.2) will be a function of n_k, n_k^2, and the cross product terms. The presence of the cross product terms will prevent factoring equation (3.2.2) into the product of one dimensional characteristic functions. Therefore, an

Eigensystem approach will be used in order to obtain s chsrscteristic function without any cross product terms.

Equation (3.2.2) is modified so as to eliminate the presence of cross product terms in the exponent. First, the $\underline{N}$ vector is defined to be

$$\underline{N} = \underline{D}\ \underline{V} \tag{3.2.3}$$

where

$\underline{N} \sim N(\underline{O},\ R_k\ \underline{I})$,

$\underline{V} \sim N(\underline{O},\ \underline{I})$,

and

$\underline{D} = \text{Diag}\ [\sqrt{R_k}]$.

Substituting (3.2.3) into (3.2.1) yields

$$y(t) = \underline{S}^H\underline{TS} + \underline{S}^H\underline{Q}^H\underline{DV} + \underline{V}^H\underline{DQS} + \underline{V}^H\underline{DPDV}. \tag{3.2.4}$$

Next, define the Gsussian random vector, $\underline{U}$, by the following transformation

$$\underline{U} = \underline{M}^H\underline{V} \tag{3.2.5}$$

where

$\underline{M}$ is a unitsry matrix whose columns are the orthonormal eigenvectors of $\underline{DPD}$,

$\underline{V} \sim N(\underline{O},\ \underline{I})$,

and

H denotes the hermitian trsnspose.

Solving for the vector $\underline{V}$ yields

$$\underline{V} = \underline{MU}. \tag{3.2.6}$$

Substituting (3.2.6) into (3.2.4) results in the filter output being expressed as

$$y(t) = \underline{S}^H\underline{TS} + \underline{S}^H\underline{Q}^H\underline{DMU} + \underline{U}^H\underline{M}^H\underline{DQS} + \underline{U}^H\underline{M}^H\underline{DPDPMU} \ . \qquad (3.2.7)$$

It can be shown that $\underline{M}^H\underline{DPDM}$ is a similarity transformation, resulting in a diagonal matrix whose nonzero elements are the eigenvalues of the $\underline{DPD}$ matrix. Let this diagonal matrix be denoted as $\underline{D}_\alpha$, where

$$\underline{D}_\alpha = \underline{M}^H\underline{DPDM} \ . \qquad (3.2.8)$$

The output of the receiver, y(t), is now given as

$$y(t) = \underline{S}^H\underline{TS} + \underline{S}^H\underline{Q}^H\underline{DMU} + \underline{U}^H\underline{M}^H\underline{DQS} + \underline{U}^H\underline{D}_\alpha\underline{U} \ . \qquad (3.2.9)$$

$$= q + \underline{R}^H\underline{U} + \underline{U}^H\underline{R} + \underline{U}^H\underline{D}_\alpha\underline{U} \qquad (3.2.10)$$

where

$$\underline{R} = \underline{M}^H\underline{DQS}, \qquad (3.2.11)$$

and

$$q = \underline{S}^H\underline{TS}. \qquad (3.2.12)$$

Equation (3.2.10) can be expanded into a series representation as follows

$$y(t) = q + \sum_{k=-K}^{K} \left(r_k^* u_k + u_k^* r_k + \alpha_k |u_k|^2\right) \ . \qquad (3.2.13)$$

It can be shown that the imaginary part of equation (3.2.13) goes to zero. This is expected since y(t) is the output of a lowpass filter and is real and even in the equivalent lowpass model.

By completing the square in equation (3.2.13) it can be rewritten as

$$y(t) = \phi_r + \sum_{k=-K}^{K} \alpha_k (u_{k_r} + \theta_{k_r})^2 + \phi_i + \sum_{k=-K}^{K} \alpha_k (u_{k_i} + \theta_{k_i})^2 \qquad (3.2.14)$$

where

$$\phi_r = q - \sum_{k=-K}^{K} \frac{r_{k_r}^2}{\alpha_k},$$

$$\phi_i = - \sum_{k=-K}^{K} \frac{r_{k_i}^2}{\alpha_k} ,$$

$$\theta_{k_r} = \frac{r_{k_r}}{\alpha_k} ,$$

and

$$\theta_{k_i} = \frac{r_{k_i}}{\alpha_k} .$$

The characteristic function of y(t), which is defined to be,

$$M_y(\nu) = E[e^{j\nu y}] \tag{3.2.15}$$

is expanded, by substituting (3.2.14) into (3.2.15), yielding

$$M_y(\nu) = E\left\{\exp[j\phi_r\nu + j\sum_k \alpha_k(u_{k_r}+\theta_{k_r})^2\nu + j\phi_i\nu + j\sum_k \alpha_k(u_{k_i}+\theta_{k_i})^2\nu]\right\}. \tag{3.2.16}$$

$$= e^{j\phi_r\nu} e^{j\phi_i\nu}\left\{\prod_{k=-K}^{K} E\left[e^{j\alpha_k(u_{k_r}+\theta_{k_r})^2\nu}\right]\right\}\left\{\prod_{k=-K}^{K} E\left[e^{j\alpha_k(u_{k_i}+\theta_{k_i})^2\nu}\right]\right\} \tag{3.2.17}$$

$$= e^{j\phi\nu} \prod_{k=-K}^{K} M_{R_k}(\nu) \prod_{k=-K}^{K} M_{I_k}(\nu) \tag{3.2.18}$$

where

$$\phi = \phi_r + \phi_i ,$$

$$R_k = \alpha_k(u_{k_r} + \theta_{k_r})^2 ,$$

$$u_{k_r} \sim N(0, 1/2),$$

$$u_{k_i} \sim N(0, 1/2),$$

and

$$I_k = \alpha_k(u_{k_i} + \theta_{k_i})^2 .$$

Evaluating the moment generating functions in equation (3.2.18) gives the following results,

$$M_{R_k}(\nu) = E\left[e^{j\alpha_k (u_{k_r} + \theta_{k_r})^2 \nu}\right]$$

$$= \frac{\exp\left[\dfrac{j\,\alpha_k\,\theta_{k_r}^2\,\nu}{(1 - j\alpha_k\,\nu)}\right]}{(1 - j\alpha_k\,\nu)^{1/2}}\,, \tag{3.2.19}$$

and

$$M_{I_k}(\nu) = E\left[e^{j\alpha_k (u_{k_i} + \theta_{k_i})^2 \nu}\right]$$

$$= \frac{\exp\left[\dfrac{j\,\alpha_k\,\theta_{k_i}^2\,\nu}{(1 - j\alpha_k\,\nu)}\right]}{(1 - j\alpha_k\,\nu)^{1/2}}\,. \tag{3.2.20}$$

By substitution of equations (3.2.19) and (3.2.20) into (3.2.18), the following expression for the characteristic function of the output, $y(t)$, is given as

$$M_y(\nu) = e^{j\phi} \prod_{k=-K}^{K} \frac{\exp\left\{\dfrac{j\,\alpha_k\,\theta_k^2\,\nu}{1 - j\,\alpha_k\,\nu}\right\}}{1 - j\,\alpha_k\,\nu} \tag{3.2.21}$$

where

$$\phi = \phi_r + \phi_i\,,$$

and

$$\theta_k^2 = \theta_{k_r}^2 + \theta_{k_i}^2\,.$$

3.3 Probability Density Function for Noise Only

The probability density function of the output, $y(t)$, is obtained by taking the Fourier transform of the characteristic function (3.2.21)

of y(t). In considering the case when only noise is incident on the receiver, equation (3.2.21) reduces to

$$M_y(\nu) = \prod_{k=-K}^{K} \left\{\frac{1}{1 - j\,\alpha_k\,\nu}\right\} . \tag{3.3.1}$$

For the case of distinct eigenvalues of the DPD matrix, equation (3.3.1) may be expressed in terms of a partial fraction expansion

$$M_y(\nu) = \sum_{k=-K}^{K} \frac{\kappa_k}{1 - j\,\alpha_k\,\nu} \tag{3.3.2}$$

where

$$\kappa_k = (1 - j\,\alpha_k\,\nu)\, M_y(\nu)\Big|_{j\nu=1/\alpha_k}$$

$$= \prod_{\substack{i=-K \\ i\neq k}}^{K} \frac{1}{1 - \alpha_i/\alpha_k} , \tag{3.3.3}$$

and

α_i is the i^{th} eigenvalue of the DPD matrix.

The probability density function of the output, y(t), for the case of noise only is the Fourier transform of equation (3.3.2), expressed as

$$P_Y(y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} M_y(\nu)\, e^{-j\nu y}\, d\nu \tag{3.3.4}$$

where

$M_y(\nu)$ is given by (3.3.2)

By applying residue theory, it can be shown that equation (3.3.4) is equivalent to

$$P_Y(y) = \sum_{k=-K}^{K} \frac{\kappa_k}{\alpha_k}\, e^{-y/\alpha_k} . \tag{3.3.5}$$

It is straightforward to show that for equation (3.3.5) the following holds

$$\int_0^{\infty} P_Y(\alpha)\ d\alpha = \sum_{k=-K}^{K} \kappa_k = 1. \tag{3.3.6}$$

For the case of noise only, equation (3.2.12) reduces to the following

$$y_n(t) = \sum_{k=-K}^{K} \alpha_k\ |u_k|^2 \tag{3.3.7}$$

where

$y_n(t)$ is the filter output with only noise present,

α_k is the k^{th} eigenvalue of the DPD matrix,

and

$u_k \sim N(0,\ 1)$ and is complex.

It follows that the mean and variance of the output for the case of noise only are

$$E[y_n(t)] = \sum_{k=-K}^{K} \alpha_k \tag{3.3.8}$$

and

$$VAR[y_n(t)] = \sum_{k=-K} \alpha_k^2\ . \tag{3.3.9}$$

3.4 Probability Density Function for Signal and Noise

The probability density function of the output, y(t), is the Fourier transform of the characteristic function (3.2.18) given by

$$P_Y(y) = \int_{-\infty}^{\infty} M_y(\nu) e^{-j\nu y}\ \frac{d\nu}{2\pi}$$

$$= \int_{-\infty}^{\infty} e^{-j\nu y} e^{j\phi y} \prod_{k=-K}^{K} \frac{\exp\left\{\dfrac{j\ \alpha_k\ \theta_k^2\ \nu}{1 - j\ \alpha_k\ \nu}\right\}}{1 - j\ \alpha_k\ \nu}\ \frac{d\nu}{2\pi}\ . \tag{3.4.1}$$

The probability density function is real and even, therefore equation (3.4.1) can be rewritten as

$$P_Y(y) = \int_0^\infty \exp\left\{-\sum_{k=-K}^{K} \frac{|r_k|^2 \, \nu^2}{(1 + \alpha_k^2 \, \nu^2)}\right\} \cdot$$

$$\cos\left[q\nu - y\nu - \sum_{k=-K}^{K}\left[\frac{|r_k|^2 \, \alpha_k \, \nu^3}{(1 + \alpha_k^2 \, \nu^2)} - \tan^{-1}(\alpha_k \nu)\right]\right]\frac{d\nu}{2\pi} \, . \tag{3.4.2}$$

For the case of signal and noise present, equation (3.2.12) is

$$y(t) = q + \sum_{k=-K}^{K} \left(r_k^* u_k + r_k u_k^* + \alpha_k \, |u_k|^2\right) \tag{3.4.3}$$

where

$y(t)$ is the output with both signal and noise present,

q is the signal portion of the output

r_k is the k^{th} element of the $\underline{M}^H\underline{DQS}$ vector,

α_k is the k^{th} eigenvalue of $\underline{DPD}$,

and

$u_k \sim N(0,1)$ and is complex.

It follows that the mean and variance of the output when both signal and noise are present are

$$E[y(t)] = q + \sum_{k=-K}^{K} \alpha_k \tag{3.4.4}$$

and

$$VAR[y(t)] = 2 \sum_{k=-K}^{K} |r_k|^2 + \sum_{k=-K}^{K} \alpha_k^2 \, . \tag{3.4.5}$$

3.5 Determination of Threshold

Detection is the occurrence of the output voltage, $y(t)$, exceeding a given threshold voltage, V_t. This threshold voltage is determined by a user specified false alarm rate (FAR). The FAR is the

average number of times per second that the output exceeds threshold with only noise incident on the receiver. The determination of the threshold voltage for an arbitrary FAR will be the subject of this section.

An approximation to the probability of a false alarm, P_f, is defined as the ratio of the number of false alarms per second, (FAR), to the number of independent opportunities for the output noise process to exceed threshold [2]. This relation is given by equations (3.5.1) and (3.5.2),

$$P_f = FAR\left(\frac{1}{2\,B_{\ell p}} + \frac{1}{B_{RF}}\right) , \tag{3.5.1}$$

which reduces to

$$P_f = \frac{FAR}{2\,B_{\ell p}} \tag{3.5.2}$$

when $B_{RF} >> B_{\ell p}$,

where

FAR is the specified false alarm rate,

$B_{\ell p}$ is the noise bandwidth of the lowpass post filter,

and

B_{RF} is the noise bandwidth of the RF prefilter.

The probability of a false alarm may be expressed as

$$P_f = 1 - P \tag{3.5.3}$$

where

P_f is the probability of a false alarm,

and

P is the probability of not exceeding threshold for the noise only case.

The probability of the output not exceeding threshold for the case of noise only is defined as

$$p = \int_0^{V_t} \text{pdf } y_n(t)\ dy$$

$$= \int_0^{V_t} \sum_{k=-K}^{K} \frac{\kappa_k}{\alpha_k} e^{-y/\alpha_k}\ dy$$

$$= \sum_{k=-K}^{K} \kappa_k (1 - e^{-V_t/\alpha_k}) \ . \tag{3.5.4}$$

Substitution of (3.5.1) and (3.5.4) into (3.5.3) yields the following expression,

$$\text{FAR}\left(\frac{1}{2\, B_{\ell p}} + \frac{1}{B_{RF}}\right) = 1 - \sum_{k=-K}^{K} \kappa_k (1 - e^{-V_t/\alpha_k}) \ ,$$

which is equivalent to

$$\text{FAR}\left(\frac{1}{2\, B_{\ell p}} + \frac{1}{B_{RF}}\right) = \sum_{k=-K}^{K} \kappa_k\, e^{-V_t/\alpha_k} \ , \tag{3.5.5}$$

which can be rewritten as a homogeneous function in V_t given by

$$F(V_t) = \text{FAR}\left(\frac{1}{2\, B_{\ell p}} + \frac{1}{B_{RF}}\right) - \sum_{k=-K}^{K} \kappa_k\, e^{-V_t/\alpha_k} \tag{3.5.6}$$

$$= 0.$$

Equation (3.5.6) is a transcendental equation in V_t, which can be solved easily using numerical techniques.

3.6 Probability of Detection

The probability of detecting a signal incident upon the receiver is developed in this section. The probability of detection, defined as $P_d(V_t)$, is a function of the previously defined threshold voltage, V_t, and is expressed as

$$P_d(V_t) = P_Y(y \geq V_t) = 1 - P_y(y < V_t) \tag{3.6.1}$$

where

y is the output of the receiver,

V_t is the threshold voltage associated with a specific FAR,

and

$P_Y(\cdot)$ is the probability density function for both signal and noise incident on the receiver.

The probability density function of the output, when both signal and noise are present was previously given as

$$P_Y(y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e(\nu) \cos [y\nu + g(\nu)] \, d\nu \tag{3.6.2}$$

where

$$e(\nu) = \frac{\exp\left\{- \sum_{k=-K}^{K} \left[\frac{|r_k|^2 \nu^2}{(1 + \alpha_k^2 \nu^2)}\right]\right\}}{\prod_{k=-K}^{K} (1 + \alpha_k^2 \nu^2)^{1/2}}, \tag{3.6.3}$$

and

$$g(\nu) = -q\nu + \sum_{k=-K}^{K} \left[\frac{|r_k|^2 \alpha_k \nu^3}{(1 + \alpha_k^2 \nu^2)} - \tan^{-1} (\alpha_k \nu)\right]. \tag{3.6.4}$$

The probability of detection, $P_d(V_t)$, is given to be

$$P_d(V_t) = 1 - P_Y(y < V_t)$$

$$= 1 - \int_{-\infty}^{0} P_Y(\alpha) \, d\alpha - \int_{0}^{V_t} P_Y(\alpha) \, d\alpha . \tag{3.6.5}$$

Consider the first integral in equation (3.6.5). It is given as

$$\int_{-\infty}^{0} P_Y(\alpha)\, d\alpha = \frac{1}{2\pi} \int_{\infty}^{0} \int_{-\infty}^{\infty} e(\nu) \cos[y\nu + g(\nu)] d\nu dy$$

$$= \int_{-\infty}^{\infty} e(\nu) \left\{ \int_{-\infty}^{0} \frac{\cos[y\nu + g(\nu)]}{2\pi} dy \right\} d\nu \qquad (3.6.6)$$

Equation (3.6.6) can be reduced by using the results of Appendix A which show that

$$\int_{-\infty}^{0} \cos \frac{[y\nu + g(\nu)]}{2\pi} dy = \frac{1}{2}\, \delta(\nu) + \frac{\sin[g(\nu)]}{2\pi\ \nu}\ . \qquad (3.6.7)$$

Substitution of equation (3.6.7) into (3.6.6) reduces the first integral in equation (3.6.5) to

$$\int_{-\infty}^{0} P_Y(\alpha)\, d\alpha = \int_{-\infty}^{\infty} e(\nu) \left(\frac{1}{2}\, \delta(\nu) + \frac{\sin g(\nu)}{2\pi}\right) d\nu$$

$$= \frac{1}{2} \int_{-\infty}^{\infty} e(\nu)\, \delta(\nu)\, d\nu + \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{e(\nu)\, \sin[g(\nu)]}{\nu} d\nu$$

$$= \frac{1}{2}\, e(0) + \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{e(\nu)\, \sin[g(\nu)]}{\nu} d\nu$$

$$= \frac{1}{2} + \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{e(\nu)\, \sin[g(\nu)]}{\nu} d\nu\ . \qquad (3.6.8)$$

The second integral in equation (3.6.5) is expanded by interchanging the order of integration and integrating over y, thus yielding

$$\int_{0}^{V_t} P_Y(\alpha)\, d\alpha = \int_{0}^{V_t} \frac{1}{2\pi} \int_{-\infty}^{\infty} e(\nu) \cos[y\nu + g(\nu)]\, d\nu dy$$

(3.6.9)

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\frac{e(\nu)\{\sin[V_t\nu + q(\nu)] - \sin[q(\nu)]\}}{\nu}\right] d\nu.$$

Finally, substitution of equations (3.6.8) and (3.6.9) into equation (3.6.5) yields the following expression for the probability of detection,

$$P_d(V_t) = 1 - \left[\frac{1}{2} + \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e(\nu)\ \sin[g(\nu)]\ d\nu}{\nu}\right]$$

$$- \left[\frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e(\nu)\{\sin[V_t\nu + g(\nu)] - \sin[g(\nu)]\}}{\nu}\,d\nu\right]$$

$$= \frac{1}{2} - \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e(\nu)\ \sin[V_t\nu + g(\nu)]\ d\nu}{\nu}$$

$$= \frac{1}{2} - \frac{1}{\pi}\int_{0}^{\infty} \frac{e(\nu)\ \sin[V_t\nu + g(\nu)]\ d\nu}{\nu} \tag{3.6.10}$$

where

$$e(\nu) = \frac{\exp\left[-\sum\limits_{k=-K}^{K} \dfrac{|r_k|^2\ \nu^2}{(1 + \alpha_k^2\ \nu^2)}\right]}{\nu\ \prod\limits_{k=-K}^{K} (1 + \alpha_k^2\ \nu^2)^{1/2}}\ k$$

and

$$g(\nu) = -q\nu + \sum_{k=-K}^{K} \left[\frac{|r_k|^2\ \alpha_k\ \nu^3}{(1 + \alpha_k^2\ \nu^2)} - \tan^{-1}(\alpha_k\nu)\right] .$$

CHAPTER 4

THE COMPUTER ANALYSIS

4.1 Introduction

A computer program, implemented on a VAX 11/750, is used to evaluate the previously defined equations in order to examine the probability of detection for the frequency compressive receiver. This chapter presents an overview of the computer program, examines the operation of the FCR through graphical output, and compares the performance of the FCR with the results that were obtained by modeling the FCR with a different approach [2]. Optimization of the FCR by varying dispersion time and/or prefilter bandwidth is presented. Given an optimized receiver, the performance is examined with and without lowpass post filtering.

4.2 Computer Program

The computer program is written in FORTRAN and designed to be user friendly and menu driven. The menus allow the user to implement different parameter configurations for the FCR. The following discussion presents the menu structure of the computer program.

First, the user is prompted for the parameters of the scanning local oscillator. These parameters are the compression bandwidth, dispersion time, and number of sample data points. The user is free to select the desired compression bandwidth and number of sample data points. The program then computes the maximum allowable dispersion time (4.2.1) so that the sampling theorem is not violated, thus eliminating the possibility of aliasing. It can be shown that the

following inequality (4.2.1) will prevent the sampled signal spectrum, which is periodic, from overlapping, i.e., no aliasing,

$$\frac{N}{4\,\beta_\omega} \geq \tau_{max} \qquad (4.2.1)$$

where

N is the number of sample data points,

β_ω is the compression bandwidth,

and

τ_{max} is the maximum dispersion time.

Figure (4.2-1a) is the output of the scanning local oscillator, in the time domain where its frequency is a linear function of time, with equation (4.2.1) satisfied. The parameters of the SLO are a compression bandwidth of 5MHz, a dispersion time of 40μs and 1024 sample data points.

The sampling frequency in Figure (4.2-1a) is 12.8 MHz, and the Nyquist frequency is 6.4 MHz. Since the Nyquist frequency is larger than the compression bandwidth no folding of frequencies occur. Thus, the output in Figure (4.2-1a) shows no evidence of aliasing. In an attempt to reduce computation time later in the program, the number of sampled data points was reduced from 1024 to 256. This reduction in the number of sampled data points clearly violates equation (4.2.1). Figure (4.2-1b) is the output of the scanning local oscillator with the number of sampled data points equaling 256.

The sampling frequency for the case in Figure (4.2-1b) is 3.2 MHz, and the Nyquist frequency is 1.6 MHz. It follows that folding of frequencies occur at integer multiples of the Nyquist frequency, namely, 1.6, 3.2, and 4.8 MHz respectively. This is apparent from Figure (4.2-1b). Thus, aliasing is present in Figure (2.4-1b). The program uses equation (4.2.1) as the default value for the dispersion time, τ.

SLO OUTPUT WITH EQUATION (4.2.1) SATISFIED

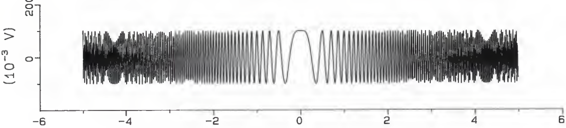


FIGURE 4.2-1a: SLO OUTPUT (MHz)

SLO OUTPUT WITH EQUATION (4.2.1) NOT SATISFIED

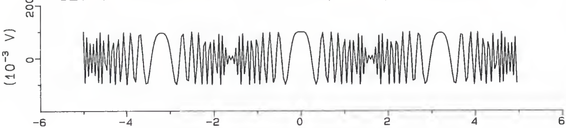


FIGURE 4.2-1b: SLO OUTPUT (MHz)

After the SLO menu, the user is prompted for the type of lowpass signal incident on the receiver. Options include a trapezoidal pulse, Gaussian pulse, or data from an external file. There is also an option to modulate the input pulse. This results in the observation of output prior to or after time $\tau/2$ seconds, as explained in section 2.2.

The next step in the program is to define the type of prefilter and its related parameters. The prefilter may be either compressive or noncompressive. The compressive filter is the default and it is automatically matched to the SLO as explained previously. Non-compressive filters are also available to the user. They include Butterworth, Chebyshev, Linear Phase, and Surface Acoustic Wave (SAW) filters.

Following the prefilter definition, the user is prompted for false alarm rate (FAR), and the lowpass noise model bandwidth, B_n. The false alarm rate is arbitrarily set to one false alarm per second. The value of B_n must be large enough so that the noise process is accurately represented. The next section discusses the effect of varying the value of B_n on the probability of detection. From this, a default value of B_n is given.

Lastly, the post filter and its related parameters are defined. The post filter may be any of the previously defined non-compressive filters as well as the option of no post filtering.

The operation of the FCR is now investigated through the use of graphical outputs. The FCR considered has a compression bandwidth of 5 MHz, an RF 3-dB frequency of 1.8 MHz, and a dispersion time of 40

μs. The signal incident on the FCR is a 20 μs pulse with a 5 μs rise time and a constant amplitude of 100 millivolts. The incident signal (equivalent lowpass), described previously, is shown in Figure (4.2-2).

The noise density, N_o, is a function of the signal-to-noise density ratio in decibels and is given as

$$N_o = \frac{A^2}{2(\mathrm{SNDR})} \tag{4.2.2}$$

where

N_o is the noise density,

A is the pulse amplitude,

and

SNDR is the signal-to-noise density ratio.

By changing the value of the SNDR, equation (4.2.2) effectively modifies the noise model so as to make it appear that pulses with different amounts of signal energy are incident, despite the constant pulse amplitude of 100 millivolts.

Figure (4.2-3) is the output of the frequency compressive receiver plotted on a time calibrated axis. Its form is that of the magnitude of a sinc function with minimal sidelobes. The dispersion time is 40 μs which is also the time of the output. This is expected since the pulse was not modulated.

Figure (4.2-4) is the output of the frequency compressive receiver plotted on a frequency calibrated axis. The frequency domain output is indicative of a lowpass pulse incident with no modulation.

Figure (4.2-5a) shows the filter output when two pulses, modulated at $\pm$ 100 kHz respectively, are incident on the reciever simultaneously. Figure (4.2-5b) shows the filter output when two

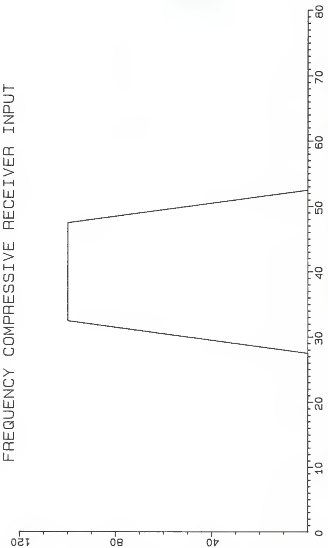


FIGURE 4.2-2: TIME DOMAIN INPUT (MICROSECONDS)

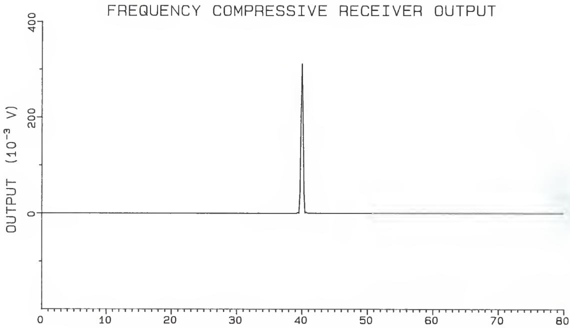


FIGURE 4.2-3: TIME DOMAIN OUTPUT (MICROSECONDS)

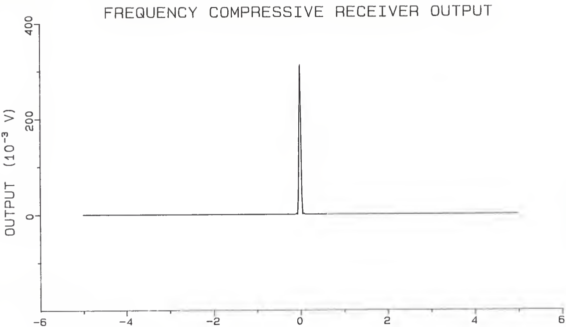


FIGURE 4.2-4: FREQUENCY DOMAIN OUTPUT (MEGAHERTZ)

FCR OUTPUT RESOLUTION AT OFFSET OF 100kHz

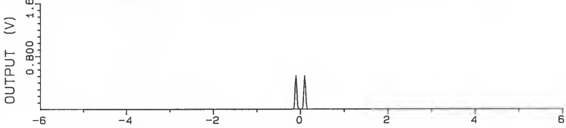


FIGURE 4.2-5a: FREQUENCY DOMAIN OUTPUT (MHz)

FCR OUTPUT RESOLUTION AT OFFSET OF 50kHz

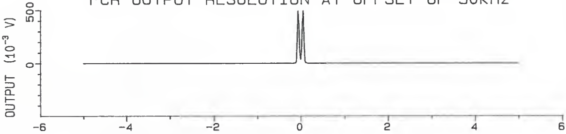


FIGURE 4.2-5b: FREQUENCY DOMAIN OUTPUT (MHz)

incident pulses are modulated at $\pm$ 50 kHz. Recalling that the resolution of the FCR is approximately $1/\tau$ to $2/\tau$, assuming certain conditions, then a dispersion time of 40 μs results in resolution of 25 kHz to 50 kHz. This agrees with the output displayed in Figures (4.2-5a and b).

Figure (4.2-6) shows the output of the FCR when five pulses are incident simultaneously. The pulses are modulated at offset frequencies of 0, $\pm$ 1 MHz, and $\pm$ 2 MHz, respectively. The attenuation of the output is a result of the 1.8 MHz RF 3-dB bandwidth.

Figure (4.2-7) shows the effect on the FCR output when the input pulse width is varied. Longer pulse widths result in highly resolvable output. This is expected since the Fourier transform of a continuous wave signal is an impulse. Smaller pulse widths result in outputs that are more spread out, since their Fourier transforms are also more spread out.

The probability density function for the case of noise only is shown in Figure (4.2-8). The noise pdf resembles that of an exponential distribution.

The probability density function for the case of signal plus noise is shown in Figure (4.2-9).

Figure (4.2-10) shows the output of the receiver when a noncompressive prefilter is used, thus modeling the conventional spectrum analyzer type intercept receiver. A 4-pole Butterworth filter with a 3-dB bandwidth of 1.8 MHz was used in place of the compressive filter.

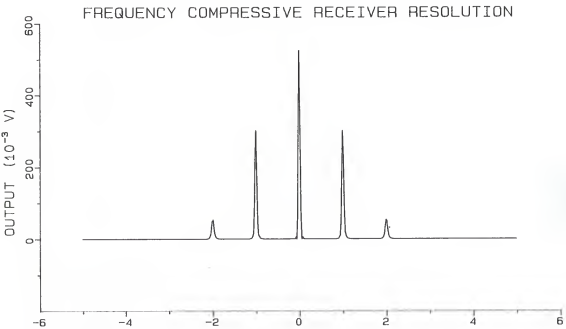


FIGURE 4.2-6: FREQUENCY DOMAIN OUTPUT (MEGAHERTZ)

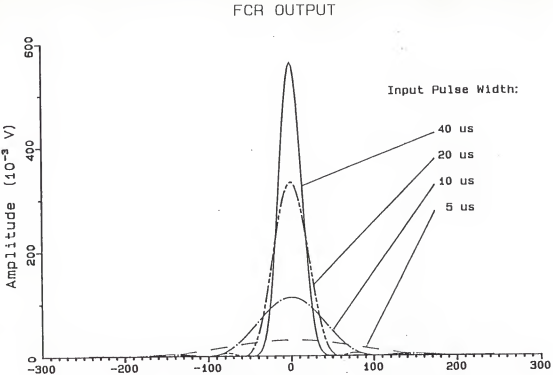


FIGURE 4.2-7: DIFFERENT INCIDENT PULSE WIDTHS (kHz)

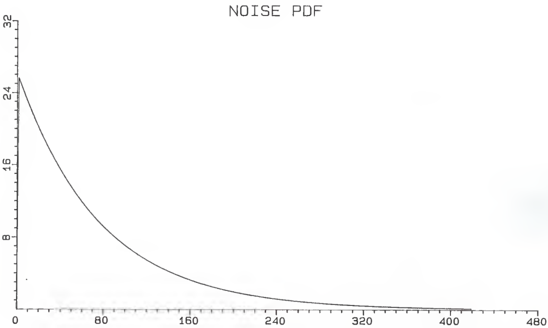


FIGURE 4.2-8: PDF OF THE NOISE OUTPUT (10^{-3} VOLTS)

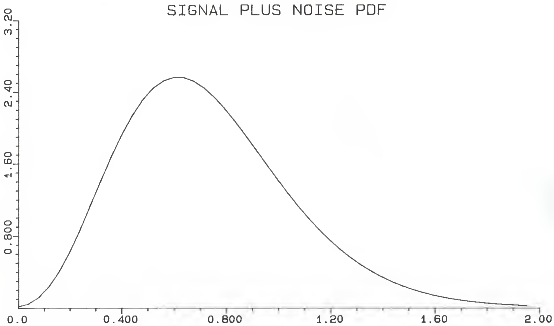


FIGURE 4.2-9: PDF OF THE SIGNAL PLUS NOISE OUTPUT (VOLTS)

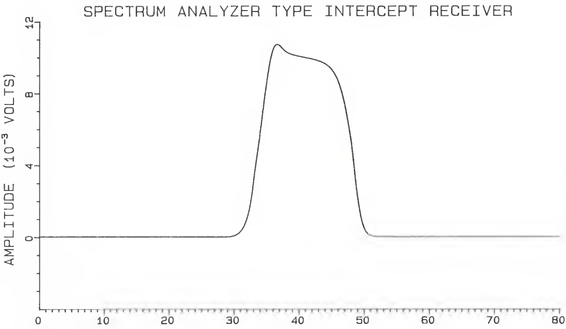


FIGURE 4.2-10: NONCOMPRESSIVE FILTER OUTPUT (MICROSECONDS)

4.3 Computer Analysis

This section utilizes the previously defined computer program to investigate the performance of the FCR under optimal conditions. In order to evaluate the probability of detection correctly, the noise process must be represented accurately. This section begins by determining an appropriate value for the noise model bandwidth, B_n. The FCR can be optimized, for a given pulse width, by modifying the filter dispersion time, τ, or the RF prefilter bandwidth, or possibly both. Optimization is investigated by varying either the dispersion time, τ, or the RF prefilter bandwidth, while the other remains fixed. The performance of the FCR is compared to results obtained previously [2], where the FCR was modeled in a different manner. The effect of post filtering the output of the FCR on the probability of detection is also considered.

It was noted previously that the mixing operation resulted in a shift in frequency of the noise power spectrum at the output of the mixer. Since the noise process incident on the receiver was assumed white, a shift in frequency still resulted in a white noise process. This resulted in an argument that omitted the effect of the SLO and defined the noise process incident on the dispersive filter as being band- limited and white. A question that arose was how wide should the bandwidth of the noise model be in order to accurately represent the noise process. The usual way is to select the noise model bandwidth, B_n, to be just a little wider than the equivalent noise bandwidth of the filter, NF3. The noise bandwidth of the dispersive filter is defined as

$$NF3 = \frac{\int_0^{\infty} |\tilde{H}(f)|^2 \, df}{|\tilde{H}(0)|^2} , \qquad (4.3.1)$$

which can be shown to be equal to

$$NF3 = \frac{F3}{2} \sqrt{\frac{\pi}{\ln_e 2}} . \qquad (4.3.2)$$

Recalling that the magnitude responae of the dispersive filter was Gaussian, one might intuitively suggest a value of B_n approximately 1.5 to 2.0 times the equivalent noise bandwidth, NF3, since Gaussian filters do not have sharp 3-dB cutoff frequencies.

Figure (4.3-1) shows the effect of varying the noise model bandwidth, B_n, on the probability of detection. The FCR used to generate this plot has an RF-3dB frequency of 170 MHz, which is an 85 MHz low-pass 3dB frequency, and an equivalent noise bandwidth of 90.45 MHz. For B_n equal to 50 MHz the receiver's performance is better than optimum. This is obviously an incorrect value for B_n. For B_n equal to 100 MHz, we observe realistic detection probabilities, slightly less than optimal, but it ia still not large enough when considering the tails of a Gaussian magnitude response.

The detection probabilitiea for B_n equal to 250 and 500 MHz are identical. Thia implies that the detection probabilities converge as B_n tenda to infinity. Thia notion makes sense intuitively. We will eventually be outside the pasa band of the dispersive filter and thus additional noise power will not be realized at the output of the filter. Based on Figure (4.3-1), the noise model bandwidth, B_n, used for the remainder of this chapter ia 500 MHz.

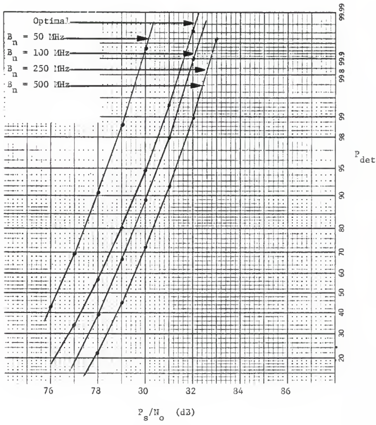


Figure 4.3-1: P_{det} for Different Noise Model Bandwidths

Given an accurate noise model representation, we can now begin to analyze the performance of the FCR. Three different lowpass incident signals are considered. They are pulsea of widtha 0.25, 0.50, 1.0 μs respectively. The FCR examined is one with a 500 MHz compression bandwidth, a 170 MHz RF 3dB bandwidth, and a dispersion time, τ, such that the receiver is optimized for the previously mentioned three cases. Resolution of the FCR, for the above cases, is determined after the observation of the optimal dispersion time.

Figures (4.3-2,3, and 4) show the probability of detection versus filter dispersion time at signal-to-noise denaity ratioa of 81, 78, and 75 dB respectively. The following table summarizes the optimal dispersion time and resolution for the three casea considered.

Case	Pulse Widths	Dispersion Time	Resolution
1	0.25 μs	0.45 μs	2.2 MHz
2	0.50 μa	0.90 μs	1.1 MHz
3	1.00 μs	1.80 μs	555.5 kHz

An interesting point is that the FCR can also be optimized by varying the RF prefilter bandwidth while the dispersion time remains fixed. From Figure (4.3-2) it is noted that a dispersion time of 0.45 μs optimizes the receiver's performance for a pulsewidth of 0.25 μs. The detection probability for this is observed to be 0.92. Now, consider a 0.25 μs pulse incident on a FCR with a suboptimal dispersion time of 0.30 μs. It was observed that this filter can be optimized by setting ita RF prefilter bandwidth to 240 MHz. The probability of detection for this case is also 0.92. Similar results were obtained

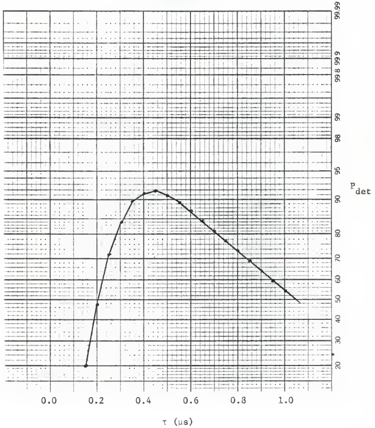


Figure 4.3-2: Optimization of Filter Dispersion Time for a 0.25 μs Pulse at P_s/N_o = 81 dB.

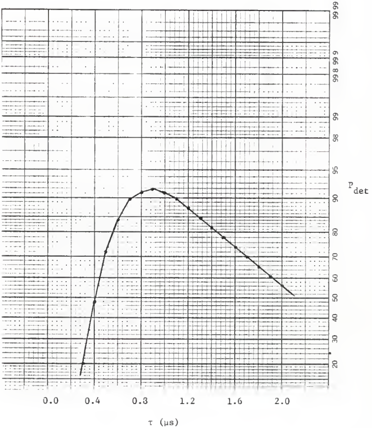


Figure 4.3-3: Optimization of Filter Dispersion Time for a 0.50 μs Pulse at P_s/N_o = 78 dB.

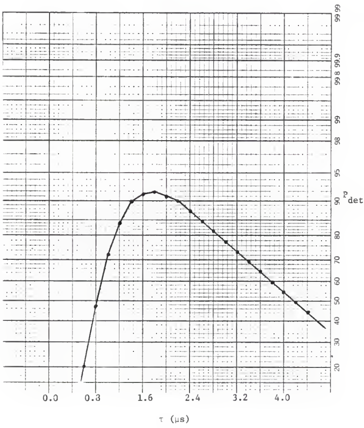


Figure 4.3-4: Optimization of Filter Dispersion Time for a 1.0 μs Pulse at P_s/N_o = 75 dB.

for a suboptimal dispersion time of 0.6 μs. It was observed that this filter can be optimized by setting its RF prefilter bandwidth to 120 MHz.

At this point it would seem appropriate to examine the relationship between dispersion time and RF bandwidth in terms of filter optimization. Figure 4.3-5 is a diagram of the RF model of the FCR which shows the relationship between dispersion time and RF bandwidth. The variables q_1, x_1, and f_{3_1} are the SLO output, signal incident on the dispersive filter, and the RF bandwidth respectively, for case number one. For case two, the variables are q_2, x_2, and f_{3_2}. Case number one represents an optimal parameter configuration, i.e. full compression is realized, thus maximizing the signal energy at the output of the dispersive filter. If the RF bandwidth in case one were changed from f_{3_1} to f_{3_2}, this would represent the filter operating under less than optimal conditions. By reducing the RF bandwidth, a portion of the down-chirp signal, incident on the dispersive filter, does not realize any compression, assuming an ideal filter. This results in less than maximum signal energy at the output. Optimization can be realized by increasing the filter dispersion, or equivalently, reducing the scan rate. This results in the SLO operating as given by q_2 rather than q_1 in the Figure. This causes a change in the slope of the down-chirp signal from that given by x_1 to that of x_2. Now the filter is operating under an optimal parameter configuration (i.e., the entire pulse is compressed).

Consider the second case, which consists of signals x_2, q_2 and an RF bandwidth of f_{3_2}. If f_{3_2} were increased to f_{3_1} then the filter would not be optimized. The filter would not be realizing any more

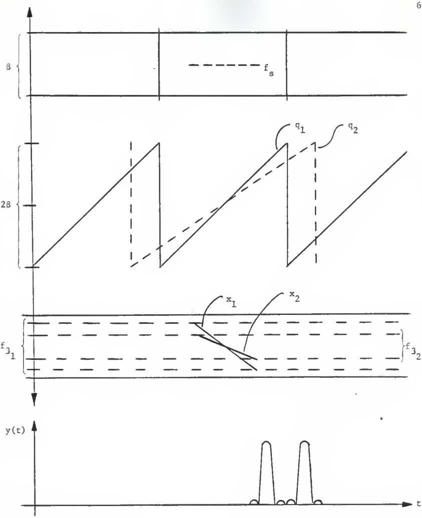


Figure 4.3-5: Optimization of the FCR.

compression, but rather just adding noise power at the output of the dispersive filter. Optimization in this situation can be realized by a reduction in the dispersion time, or equivalently, an increase in the acanning rate. The increased scanning rate, results in changing the form of the down-chirp signal incident on the dispersive filter from that of x_2 to that of x_1. The filter is now optimized.

Figures (4.3-6,7, and 8) show the performance of the FCR in terms of its probability of detection as a function signal-to-noise density ratio in dB. Detection probabilities are also given for the optimal receiver detecting a signal of unknown phase [2]. The important point in these plots is that these probabilities are identical to results obtained by modeling the FCR in a different manner [2].

Figures (4.3-9,10, and 11) show the effect of post filtering the output of the FCR on the probability of detection. In the first two cases, lowpaaa post filtering tends to degrade performance, where the amount of degradation is proportional to the filter's bandwidth. The last caae indicates that the post filter has essentially no effect on the probability of detection. The literature tended to support the notion that post filtering does not improve performance. This is in agreement with Figures (4.3-9,10, and 11) and thus, poat filtering should not be a design conaideration when constructing a FCR.

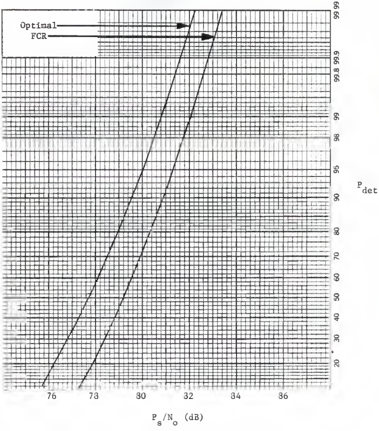


Figure 4.3-6: Receiver Performance for a Pulse Width of 0.25 μs.

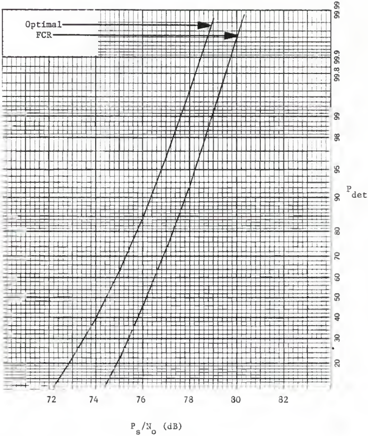


Figure 4.3-7: Receiver Performance for a Pulse Width of 0.50 μs.

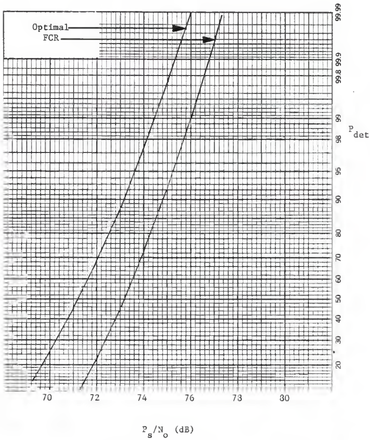


Figure 4.3-3: Receiver Performance for a Pulse Width of 1.0 μs.

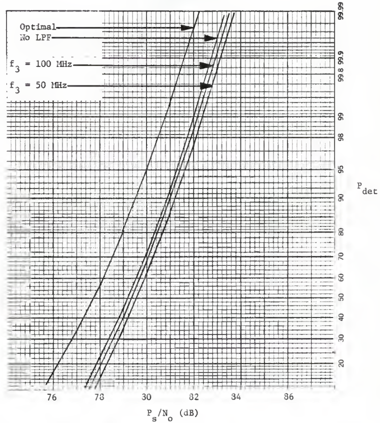


Figure 4.3-9: Performance of the FCR with Lowpass Filtering for a 0.25 μs Pulse.

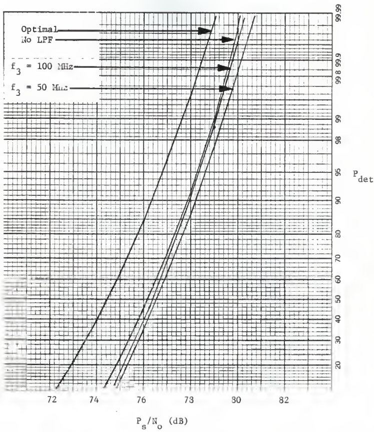


Figure 4.3-10: Performance of the FCR with Lowpass Filtering for a 0.50 μs Pulse.

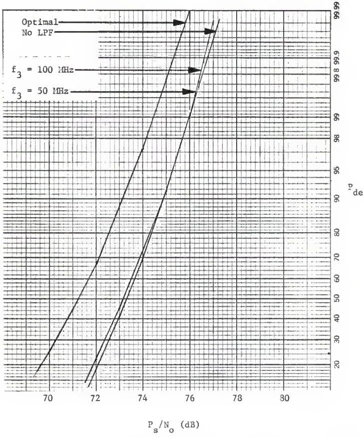


Figure 4.3-11: Performance of the FCR with Lowpass Filtering for for a 1.0 μs Pulse.

CHAPTER 5

CONCLUSION

This chapter concludes this thesis with an overall summary of interesting points observed throughout this research.

First, the frequency compressive receiver was successfully modeled by a finite series representation of the signal and noise incident on the receiver. The measures of the receiver's performance, i.e. detection probabilities, were in very close agreement with those observed from previous work [2]. This previous work did not consider post filtering and showed that the output of the square-law device for the noise had an exponential distribution and the signal plus noise had a distribution corresponding to the sum of squares of two Gaussian random variables. With such close agreement in the receiver's performances, one might accept the notion that modeling a nonstationary noise process (output of the mixer) by a stationary noise process does not compromise the integrity of the overall model.

Secondly, optimization of the frequency compressive receiver, given knowledge of the incident signal, was examined. It was shown that optimization could be achieved by varying the filter's RF bandwidth for a fixed dispersion time or by varying the filter's dispersion time for a fixed RF bandwidth. Given a filter with a fixed dispersion time, optimization may be realized by the addition of an RF prefilter with a variable 3-dB passband, thus making the filter more versatile.

Lastly, the effect of lowpass filtering the output of the frequency compressive receiver on the probability of detection was examined. Post filters considered were 2-pole and 4-pole Butterworth

filters with different cutoff frequencies. For smaller pulses incident on the receiver, post filtering resulted in a degradation of the receiver's performance. As the pulse width increased, post filtering did not alter the receiver's performance. At large values of signal-to-noise density ratio, the required performance was slightly improved. This improvement was not enough to warrant post filtering considerations in the design of the FCR.

APPENDIX A

SOLUTION TO A USEFUL INTEGRAL

This appendix presents a detailed solution of the following integral

$$\int_{-\infty}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega \,. \tag{A.1}$$

The solution to (A.1) was necessary in evaluating the probability density function for the case of both signal and noise present. The integral (A.1) is first partitioned as follows

$$\int_{-\infty}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega = \int_{-\infty}^{-\phi/t} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega + \int_{-\phi/t}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega \tag{A.2}$$

Then, define $x = (\omega + \phi/t)$ so that $dx = d\omega$ and substitute into the first integral in (A.2), thus yielding

$$\begin{aligned} \int_{-\infty}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega &= \int_{-\infty}^{0} \frac{\cos(xt)}{2\pi}\, dx + \int_{-\phi/t}^{0} \frac{\cos(\omega t + \phi)}{2r}\, d\omega \\ &= \frac{1}{2} \int_{-\infty}^{\infty} \frac{\cos(xt)}{2\pi}\, dx + \int_{-\phi/t}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega \\ &= \frac{1}{2}\, \delta(t) + \frac{\sin \phi}{2\pi t} \,. \end{aligned} \tag{A.3}$$

The result (A.3) is used in equation (3.6.7) in solving for the probability of detection.

APPENDIX B

FORTRAN SOURCE CODE

```
      INTEGER              TYPEPRE,TYPEPOST,KPOINT,NPOINT,FILTNUM

      LOGICAL              RUN

      COMPLEX              SIGNAL(0:1024),P(-31:31,-31:31),
     +                     EGVECT(-31:31,-31:31),TM(-512:512,-512:512),
     +                     Q(-31:31,-512:512),SCOEF(-512:512),RMAT(-31:31),
     +                     TEMP(-31:31)

      REAL                 FK(-31:31),RK(-31:31),NO,PLENGTH,FAR,
     +                     TAU,SCANW,BETAW,A(12),B(12),NPREF3,NPOSTF3,
     +                     PHI(0:1024),OELTAT,Y(0:1024),PREF3,POSTF3,
     +                     T,O(-31:31),EGVALS(-31:31),TO,THETAO,YS,
     +                     DELTAY,S(0:1024),PY(0:1024),S1,
     +                     YMEANN,YVARN,YMEANS,YVARS,POFN(0:1024)
      DOUBLE COMPLEX       OPO(-31:31,-31:31)

      DOUBLE PRECISION OPOR(-31:31,-31:31),OPOI(-31:31,-31:31),
     +                     EGVECR(-31:31,-31:31),
     +                     EGVECI(-31:31,-31:31),POFNO(0:1024)

      DOUBLE PRECISION OEGVALS(-31:31),KK(-31:31),VT,SUMK


      CHARACTER*9          NAME1,NAME2,NAME3,NAME4,NAME5

      COMMON /PRE/TO,THETAO

975   S1 = 0.0
      PI = 3.1415926
      KPOINT = 63
      RUN = .TRUE.
      TPOET = 1.0
      POET = 0.0
C
C  Oefine the parameters of the scanning local oscillator
C
      CALL SLO(SCANW,BETAW,PLENGTH,TAU,OELTAT,NPOINT)
C
C  Oefine the signal that is incident on the receiver
C
      CALL INCIOENT(SIGNAL,S,RUN,NO,PLENGTH,NPOINT,OELTAF)

      IF (.NOT.RUN) THEN
         GOTO 999
      ENOIF
C
C  Run the signal through the mixer
C
      CALL SCALE(SIGNAL,PHI,OELTAT,NPOINT,SCANW,PLENGTH)
C     CALL DUMPYOUT(SIGNAL,NPOINT,DELTAT,YMAX,TMAX,Y)
C
C  Oefine the Prefilter, Postfilter, and the Noise parameters
C
      CALL PREOEF(A,PREF3,NPREF3,TYPEPRE,SCANW,TO,THETAO,TAU)
      CALL NOISE(FK,RK,NO,BN,FAR,NPREF3)
      CALL POSTOEF(B,POSTF3,NPOSTF3,TYPEPOST,FILTNUM)
      CALL DFT(SIGNAL,NPOINT,0)
```

```
      SCOEF(0) = SIGNAL(0)
      DO I = 1,NPOINT/2-1
         SCOEF(I)  = SIGNAL(I)
         SCOEF(-I) = SIGNAL(NPOINT-I)
      END DO
C
C  Send the Signal through the filter on the front end
C
      CALL FILTER(PREF3,TYPEPRE,A,NPOINT,SIGNAL,SCANW,PLENGTH)
      CALL DFT(SIGNAL,NPOINT,1)
C
C  Send the Signal through the Square Law Device
C
      CALL SQUARE(SIGNAL,NPOINT)

      IF (FILTNUM.EQ.6) THEN
         GOTO 135
      END IF

      CALL DFT(SIGNAL,NPOINT,0)
C
C  Send the Signal through the Low Pass Post filter
C
      CALL FILTER(POSTF3,TYPEPOST,B,NPOINT,SIGNAL,SCANW,PLENGTH)
      CALL DFT(SIGNAL,NPOINT,1)
C
C  Write the Time Domain output to an external file
C
135   CALL DUMPYOUT(SIGNAL,NPOINT,DELTAT,YMAX,TMAX,Y)
      CALL TIMEIND(NINDEP,NPOSTF3,NPREF3,TMAX,NI,FILTNUM,DELTAT)

      DO WHILE (NI.LE.NPOINT-1)

            IF (Y(NI).GT.YMAX/10) THEN
            TYPE *,' '
            TYPE *,'  OBS NUMBER:',NI

            T = NI*DELTAT

            TYPE *,'  TIME OF OBS:',T
            TYPE *,' '

               YS=Y(NI)
               CALL PMATRIX(KPOINT,P,FK,PREF3,TYPEPRE,POSTF3,
     +                      TYPEPOST,A,B,SCANW,T)
      TYPE *,'P-MATRIX'
               CALL DIAGD(KPOINT,D,RK)
      TYPE *,'DIAGD'
               CALL DPOX(KPOINT,DPD,D,P)
      TYPE *,'DPDX'
               CALL EIGEN(KPOINT,DPDR,DPDI,DPD,DEGVALS,EGVECT,
     +                    EGVALS,EGVECR,EGVECI)
      TYPE *,'EIGEN'
               CALL QANOR(KPOINT,NPOINT,D,Q,SCOEF,RMAT,R,EGVECT,
     +                    PREF3,TYPEPRE,A,POSTF3,TYPEPOST,
     +                    B,T,FK,SCANW,PLENGTH)
      TYPE *,'QANDR'

               CALL STATS(YMEANN,YVARN,YMEANS,YVARS,KPOINT,EGVALS,
     +                    RMAT,YS)
```

```
                DELTAY = ( YMEANN + 10.*SQRT(YVARN) )/NPOINT
                CALL NOISEPDF(KPOINT,NPOINT,DEGVALS,KK,PDFND,
     +                    DELTAY,EGVALS,PDFN)
                CALL QUERY('COMPUTE SIGNAL + NOISE PDF?',IANS)
                IF (IANS.EQ.1) THEN
                CALL SANDNPDF(PY,YMEANS,YVARS,NAME4,S1,YS,KPOINT,EGVALS,
     +                    RMAT,YMEANN,NPOINT)
                END IF
                CALL THRESH(VT,FAR,NPOSTF3,NPREF3,KK,DEGVALS,KPOINT,
     +                    YMEANN)

                CALL PDETECT(S1,PDET,YMEANN,YVARN,KPOINT,EGVALS,
     +                    RMAT,YS,VT)

                TPDET=TPDET*(1.-PDET)

                END IF
               NI = NI + NINDEP

      END DO

      TPDET=1.-TPDET
      TYPE *,'THE TOTAL PROBABILITY OF DETECT IS ',TPDET
C     CALL DUMPINFD(..............OM

C
C  Do you want to run the program again?
C
      CALL QUERY('Do you want to PLAY again? ',IANS)

      IF (IANS.EQ.1) THEN
         GOTO 975
      END IF

      TYPE *,'THE RADIAN SCAN RATE IS',SCANW
      TYPE *,'THE COMPRESSION BW IN RADIANS IS',BETAW
      TYPE *,'THE PULSE LENGTH IS',PLENGTH
      TYPE *,'THE DISPERSION TIME IS',TAU
      TYPE *,'THE TIME INCREMENT IS',DELTAT
      TYPE *,'THE NUMBER OF POINTS IS',NPOINT
      TYPE *,'THE NOISE DENSITY IS',NO
      TYPE *,'THE SHIFT IN FREQUENCY IS',DELTAF
      TYPE *,'THE MAX OUTPUT VALUE IS',YMAX
      TYPE *,'THE TIME OF MAX OUTPUT IS',TMAX
      TYPE *,'THE PREFILTER BW IS',PREF3
      TYPE *,'THE PREFILTER NOISE BW IS',NPREF3
      TYPE *,'THE TYPE OF PREFILTER IS',TYPEPRE
      TYPE *,'TD IS',TD
      TYPE *,'THETAO IS',THETAO
      TYPE *,'THE NOISE BANDWIDTH IS',BN
      TYPE *,'THE FALSE ALARM RATE IS',FAR
      TYPE *,'THE POST FILTER BW IS',POSTF3
      TYPE *,'THE POSTFILTER NOISE BW IS',NPOSTF3
      TYPE *,'THE TYPE OF POSTFILTER IS',TYPEPOST
      TYPE *,'THE FILTER NUMBER IS',FILTNUM

999   STOP
      END

C****************************************************************
```

```
      SUBROUTINE   INCIOENT(SIGNAL,S,RUN,NO,PLENGTH,NPOINT,DELTAF)

      INTEGER               IIN(20),IOUT(20),ERROR,PNUM
      CHARACTER*80          MENU,PAGENAME,COUT(20),CIN(20),STR(20)
      COMPLEX               SIGNAL(0:1024),TSIGNAL(0:1024)
      REAL                  ROUT(20),RIN(20),S(0:1024),NO,NFIG,
     +                      NO1,NO2,NFIG1,NFIG2,TS(0:1024)
      DOUBLE PRECISION      DOUT(20),OIN(20)
      LOGICAL               FALSE,RUN

      COMMON /SCROAT/MENU,FALSE,IOUT,COUT,ROUT,OOUT,IIN,CIN,RIN,DIN,
     +              STR,ERROR

      MENU='MENU.TXT'
      FALSE = .FALSE.
      IIN(1) = 1

100   CALL SCRGEN('SIGNAL')
      IF (IIN(1).EQ.1) THEN
         OIN(1)=PLENGTH
         OIN(2)=0.25E-6
         OIN(3)=0.0
         DIN(4)=0.325E-6
         DIN(5)=80.0E0
         OIN(6)=1.0
         OIN(7)=1.E12
         OIN(8)=0.E0

         IIN(1)=NPOINT

         CALL SCRGEN('TRAPZOIO')

         PLENGTH=OIN(1)
         PWIDTH=DIN(2)
         PRISE=OIN(3)
         PDELAY=OIN(4)
         SNDROB=OIN(5)
         NFIG=DIN(6)
         GAIN=OIN(7)
         OELTAF=OIN(8)

         NPOINT=IIN(1)

         SNOR = 10.**(SNOROB/10.)
         NO = 0.01/(2.*SNOR)
         PAMP = 0.1

         CALL TRAP(PLENGTH,PWIDTH,PRISE,POELAY,PAMP,NPOINT,SIGNAL,S)

         OELTAT=PLENGTH/NPOINT
         IF(DELTAF.NE.0.0) THEN
           CALL SHIFTF(SIGNAL,OELTAT,OELTAF,0,NPOINT)
         END IF

      ELSE
        IF (IIN(1).EQ.2) THEN

           I=1
           PNUM=2
```

```
200         IIN(1)=PNUM
            DIN(1)=PLENGTH
            DIN(2)=(0.25)*PLENGTH
            DIN(3)=(0.05)*PLENGTH
            DIN(4)=(0.35)*PLENGTH
            DIN(5)=63.0+E0
            DIN(6)=1.0
            DIN(7)=1.E12
            DIN(8)=0.0
            IDUT(1)=I

            CALL SCRGEN('MULTIPLE')

            PNUM=IIN(1)
            PLENGTH=DIN(1)
            PWIDTH=DIN(2)
            PRISE=DIN(3)
            PDELAY=DIN(4)
            SNDRDB=DIN(5)
            NFIG=DIN(6)
            GAIN=DIN(7)
            DELTAF=DIN(8)

            SNDR=10**(SNDRDB/10.)
            ND =  0.01/(2*SNDR)
            PAMP = 0.1

            CALL TRAP(PLENGTH,PWIDTH,PRISE,PDELAY,PAMP,NPOINT,
     +                TSIGNAL,TS)

            DELTAT=PLENGTH/NPOINT
            IF (DELTAF.NE.0.0) THEN
               CALL SHIFTF(TSIGNAL,DELTAT,DELTAF,0,NPOINT)
            END IF


            DO J = 0,NPOINT-1
               SIGNAL(J)=SIGNAL(J) + TSIGNAL(J)
                     S(J) = S(J) + TS(J)
            END DO

            I = I+1
            IF (I.LE.PNUM) THEN
               GOTD 200
            END IF

         ELSE
            IF (IIN(1).EQ.3) THEN
               CALL SCRGEN('GAUSS')
            ELSE
                 IF (IIN(1).EQ.4) THEN
                    CALL SCRGEN('DISKIO')
                  ELSE
                      IF (IIN(1).EQ.5) THEN
                         RUN = .FALSE.
                         RETURN
                      ELSE
                         GOTD 100
                      ENDIF
```

```
                  ENDIF
               ENOIF
            ENDIF
         ENDIF

         RETURN
         END
C*******************************************************************
         SUBROUTINE TRAP(PL,PW,PR,PD,PA,NP,SIG,S)

         COMPLEX                  SIG(0:1024)
         REAL                     S(0:1024),PL,PW,PR,PD,PA,DELTAT
         INTEGER                  CODE
         CHARACTER*9              NAME

         DELTAT=PL/NP
         I=0

         DO WHILE (I*DELTAT.LT.PD)
               SIG(I)=CMPLX(0.0,0.0)
                 S(I)=REAL(SIG(I))
                   I=I+1
         END DO

         DO WHILE (I*DELTAT.LT.PR+PD)
               SIG(I)=CMPLX((PA/PR)*(I*DELTAT-PO),0.0)
                 S(I)=REAL(SIG(I))
                   I=I+1
         END DO

         DO WHILE (I*DELTAT.LT.PD+PW)
               SIG(I)=CMPLX(PA,0.0)
                 S(I)=REAL(SIG(I))
                   I=I+1
         END DO

         DO WHILE (I*DELTAT.LT.PW+PR+PD)
               SIG(I)=CMPLX((PW+PR+PD-I*DELTAT)*(PA/PR),0.0)
                 S(I)=REAL(SIG(I))
                   I=I+1
         END DO

         DO WHILE (I*DELTAT.LT.PL)
               SIG(I)=CMPLX(0.0,0.0)
                 S(I)=REAL(SIG(I))
                   I=I+1
         END DO

         CALL SGOPEN(2,'WRITE','SIGNAL FILE ? ',NAME,'REAL',NP)
         CALL SGTRAN(2,'WRITE','REAL',S,NP)

         RETURN
         END
C*******************************************************************
         SUBROUTINE NOISE(FK,RK,NO,BN,FAR,NPREF3)
```

```
      COMMON/SCROAT/MENU,FALSE,IOUT,COUT,ROUT,OOUT,IIN,CIN,RIN,OIN,
     +              STR,ERROR

      REAL                 NUK(-31:31),GAMK(-31:31),GQR(126),RIN(20),
     +                     ROUT(20),FK(-31:31),RK(-31:31),NO,FAR,NPREF3
      CHARACTER*9          NAME
      CHARACTER*80         MENU,PAGENAME,COUT(20),CIN(20),STR(20)
      DOUBLE PRECISION     OIN(20),DOUT(20)
      LOGICAL              FALSE
      INTEGER              IIN(20),IOUT(20),ERROR

      PI = 3.1415926
      NAME = 'GQR63.OAT'
      NOBS = 126
      CALL SGOPEN(0,'READ','NOPROMPT',NAME,'REAL',NOBS)
      CALL SGTRAN(0,'READ','REAL',GQR,NOBS)


      DO I = 1,63,1
             NUK(I-32) = GQR(I)
             GAMK(I-32) = GQR(I + 63)
      END DO

      RIN(1) = (3./2.)*NPREF3
      RIN(2) = 1.0

      CALL SCRGEN('NOISE')

      BN = RIN(1)
      FAR  = RIN(2)


      DO I = -31,31,1
           FK(I) = BN*NUK(I)
           RK(I) = 2*BN*NO*GAMK(I)
      ENO DO

      RETURN

      END
C*******************************************************************

      SUBROUTINE SLO(SCANW,BETAW,PLENGTH,TAU,DELTAT,NPOINT)

      REAL                 SCANW,SCANF,BETAW,PLENGTH,OELTAT,
     +                     ROUT(20),RIN(20)
      INTEGER              IIN(20),IOUT(20),ERROR
      CHARACTER*80         MENU,COUT(20),CIN(20),STR(20)
      DOUBLE PRECISION     DOUT(20),DIN(20)
      LOGICAL              FALSE

      COMMON /SCRDAT/MENU,FALSE,IOUT,COUT,ROUT,DOUT,IIN,CIN,RIN,DIN,
     +               STR,ERROR

      MENU='MENU.TXT'
      FALSE=.FALSE.

      PI=3.1415926
```

```
      DIN(1)=500.0E6
      IIN(1)=512

      CALL SCRGEN('SCANNING')

      BETAF=DIN(1)
      NPOINT=IIN(1)
      IDUT(1) = NPOINT
      DOUT(1) = BETAF
      DOUT(2) = NPOINT/(4*BETAF)
      DIN(1)= NPOINT/(4*BETAF)

      CALL SCRGEN('SCANN')

      TAU=DIN(1)

      BETAW=BETAF*2*PI
      PLENGTH=2*TAU
      DELTAT=PLENGTH/NPOINT
      SCANF=BETAF/TAU
      SCANW=2*PI*SCANF

      RETURN

      END
C***********************************************************************

      SUBRDUTINE SCALE(SIGNAL,PHI,DELTAT,NPOINT,SCANW,PLENGTH)

      CDMPLEX                SIGNAL(0:1024)
      REAL                   PHI(0:1024),RL,IM,SCANW,PLENGTH

      PI = 3.1415926
      SCANF = SCANW/(2*PI)

      DO I = 0,NPOINT-1,1

                PHI(I) = -PI*PLENGTH*SCANF*(I*DELTAT) +
     &                   PI*SCANF*((I*DELTAT)**2)
                    RL = CDS(PHI(I))
                    IM = -SIN(PHI(I))
             SIGNAL(I) = SIGNAL(I)*CMPLX(RL,IM)

      END DO

      RETURN

      END
C***********************************************************************

      SUBRDUTINE PREDEF(A,PREF3,NPREF3,TYPE,SCANW,TO,THETAO,TAU)

      INTEGER                IIN(20),IDUT(20),ERROR,
     +                       FILTNUM,NPOLE,NRIP,TYPE
      CHARACTER*80           MENU,COUT(20),CIN(20),STR(20)
      REAL                   RDUT(20),RIN(20),A(12),NPREF3,SCANW,PREF3
      DOUBLE PRECISION       DOUT(20),DIN(20)
      LOGICAL                FALSE
```

```
      COMMON /SCRDAT/MENU,FALSE,IOUT,COUT,ROUT,DOUT,IIN,CIN,RIN,OIN,
     +               STR,ERROR

      PI =3.1415926

      DO I = 1,12,1
         A(I) = 0.0
      END DO

      IIN(1)=1
      CALL SCRGEN('PREFILT')
      IF (IIN(1).EQ.1) THEN
         DIN(1)=170.0E6
         DIN(2)=SCANW
         DIN(3)=0.0
         DIN(4)=0.0
         CALL SCRGEN('COMPRESS')
         PREF3=DIN(1)/2.
         SCANW=DIN(2)
         TO=DIN(3)
         THETAO=DIN(4)
         TYPE=0
         NPREF3=(0.5)*PREF3*SQRT(PI/ALOG(2.0))
         GOTO 600
      ELSE
         COUT(1)='NONCOMPRESSIVE RECEIVERS'
         COUT(2)='************************'
         IIN(1)=1
         DIN(1)=3.E6
         CALL SCRGEN('NONCMP')
         FILTNUM=IIN(1)
         PREF3=DIN(1)
         IF (FILTNUM.EQ.1) THEN
            IIN(1)=2
            CALL SCRGEN('POLES')
            NPOLE=IIN(1)
            TYPE=1
            NPREF3=(PREF3*PI)/(2*NPOLE*SIN(PI/(2*NPOLE)))
         ELSE
             IF (FILTNUM.EQ.2) THEN
                IIN(1)=2
                CALL SCRGEN('POLES')
                NPOLE=IIN(1)
                TYPE=1
                NPREF3=PREF3
                IIN(1)=2
                CALL SCRGEN('RIPPLE')
                NRIP=IIN(1)
             ELSE
                IF (FILTNUM.EQ.3) THEN
                   IIN(1)=2
                   CALL SCRGEN('POLES')
                   NPOLE=IIN(1)
                   TYPE=1
                   NPREF3=PREF3
                ELSE
                   IF (FILTNUM.EQ.4) THEN
                      TYPE=2
                      NPREF3=PREF3
                   ELSE
```

```
                        TYPE=3
                        NPREF3=PREF3
                     END IF
                  END IF
               END IF
            END IF
         END IF

         CALL DEFINE(FILTNUM,NPOLE,NRIP,A,NPREF3,PREF3)

   600   RETURN
         END

C******************************************************************

         SUBROUTINE POSTDEF(B,POSTF3,NPOSTF3,TYPE,FILTNUM)

         INTEGER          IIN(20),IOUT(20),ERROR,FILTNUM,NPOLE,NRIP,TYPE
         CHARACTER*80     MENU,COUT(20),CIN(20),STR(20)
         REAL             ROUT(20),RIN(20),B(12),NPOSTF3,POSTF3
         DOUBLE           PRECISION DOUT(20),DIN(20)
         LOGICAL          FALSE

         COMMON /SCRDAT/MENU,FALSE,IOUT,COUT,ROUT,DOUT,IIN,CIN,RIN,DIN,
      +                   STR,ERROR

         PI=3.1415926

         DO I = 1,12,1
            B(I) = 0.0
         END DO

         COUT(1) = 'POST FILTER DEFINITION'
         COUT(2) = '**********************'
         IIN(1)=6
         DIN(1)=1.0E6

         CALL SCRGEN('NONCMP')

         FILTNUM=IIN(1)
         POSTF3=DIN(1)

         IF (FILTNUM.EQ.1) THEN
            IIN(1)=2
            CALL SCRGEN('POLES')
            NPOLE=IIN(1)
            TYPE=1
            NPOSTF3=(POSTF3*PI)/(2*NPOLE*SIN(PI/(2*NPOLE)))
         ELSE
            IF (FILTNUM.EQ.2) THEN
               IIN(1)=2
               CALL SCRGEN('POLES')
               NPOLE=IIN(1)
               TYPE=1
               NPOSTF3=POSTF3
               IIN(1)=2
               CALL SCRGEN('RIPPLE')
               NRIP=IIN(1)
            ELSE
               IF (FILTNUM.EQ.3) THEN
```

```
                  IIN(1)=2
                  CALL SCRGEN('POLES')
                  NPOLE=IIN(1)
                  TYPE=1
                  NPOSTF3=POSTF3
               ELSE
                  IF (FILTNUM.EQ.4) THEN
                     TYPE=2
                     NPOSTF3=POSTF3
                  ELSE
                     IF (FILTNUM.EQ.5) THEN
                        TYPE=3
                        NPOSTF3=POSTF3
                     ELSE
                        TYPE = 1
                        POSTF3 = 5.E+9
                       NPOSTF3 = 5.E+9
                     END IF
                  END IF
               END IF
         END IF
   END IF

      CALL DEFINE(FILTNUM,NPOLE,NRIP,B,NPOSTF3,POSTF3)

      RETURN
      END

C*******************************************************************

      SUBROUTINE  DEFINE(FILTNUM,NPOLE,NRIP,A,NF3,F3)

      REAL                  A(12),NF3,F3
      INTEGER               FILTNUM,NPOLE,NRIP

      PI=3.1415926
      TYPE=1

      GO TO (100,200,300,600,600,390),FILTNUM
100   GO TO (110,120,130,140,150,160,170,180),NPOLE
110   A(1)=1
      A(4)=1
      A(5)=1
      GOTO 600
120   A(1)=1
      A(4)=1
      A(5)=1.41421
      A(6)=1
      GOTO 600
130   A(1)=1
      A(4)=1
      A(5)=2
      A(6)=2
      A(7)=1
      GOTO 600
140   A(1)=1
      A(4)=1
      A(5)=2.6131
      A(6)=3.4142
      A(7)=2.6131
```

```
      A(8)=1
      GOTO 600
150   A(1)=1
      A(4)=1
      A(5)=3.2361
      A(6)=5.2361
      A(7)=5.2361
      A(8)=3.2361
      A(9)=1
      GOTO 600
160   A(1)=1
      A(4)=1
      A(5)=3.8637
      A(6)=7.4641
      A(7)=9.1416
      A(8)=7.4641
      A(9)=3.8637
      A(10)=1
      GOTO 600
170   A(1)=1
      A(4)=1
      A(5)=4.494
      A(6)=10.0978
      A(7)=14.5918
      A(8)=14.5918
      A(9)=10.0978
      A(10)=4.494
      A(11)=1
      GOTO 600
180   A(1)=1
      A(4)=1
      A(5)=5.1258
      A(6)=13.1371
      A(7)=21.8462
      A(8)=25.6884
      A(9)=21.8462
      A(10)=13.1371
      A(11)=5.1258
      A(12)=1
      GOTO 600
200   GO TO(200,220,230,240,250,260,270,280),NPOLE

220   GO TO(221,222,223,224),NRIP
221   A(1)=.50062
      A(4)=.70715
      A(5)=.6442
      A(6)=1
      GOTO 600
222   A(1)=.66276
      A(4)=.74363
      A(5)=.90151
      A(6)=1
      GOTO 600
223   A(1)=.8676
      A(4)=.87765
      A(5)=1.22081
      A(6)=1
      GOTO 600
224   A(1)=.9542
      A(4)=.9553
```

```
      A(5)=1.34868
      A(6)=1
      GOTO 600
230   GOTO (231,232,233,234),NRIP
231   A(1)=.25035
      A(4)=.25035
      A(5)=.92774
      A(6)=.59706
      A(7)=1
      GOTO 600
232   A(1)=.37429
      A(4)=.37429
      A(5)=1.03303
      A(6)=.90268
      A(7)=1
      GOTO 600
233   A(1)=.61123
      A(4)=.61123
      A(5)=1.36286
      A(6)=1.39582
      A(7)=1
      GOTO 600
234   A(1)=1
      A(4)=.78718
      A(5)=1.64659
      A(6)=1.69337
      A(7)=1
      GOTO 600
240   GOTO (241,242,243,244),NRIP
241   A(1)=.1252
      A(4)=.1769
      A(5)=.4046
      A(6)=1.1689
      A(7)=.5812
      A(8)=1
      GOTO 600
242   A(1)=.1998
      A(4)=.2242
      A(5)=.636
      A(6)=1.3112
      A(7)=.9049
      A(8)=1
      GOTO 600
243   A(1)=.3782
      A(4)=.3826
      A(5)=1.1346
      A(6)=1.785
      A(7)=1.4869
      A(8)=1
      GOTO 600
244   A(1)=.5622
      A(4)=.5629
      A(5)=1.599
      A(6)=2.2936
      A(7)=1.9125
      A(8)=1
      GOTO 600
250   GOTO (251,252,253,254),NRIP
251   A(1)=.06261
      A(4)=.06261
```

```
      A(5)=.4078
      A(6)=.5488
      A(7)=1.4147
      A(8)=.5745
      A(9)=1
      GOTO 600
252   A(1)=.104
      A(4)=.104
      A(5)=.5083
      A(6)=.8820
      A(7)=1.5803
      A(8)=.9062
      A(9)=1
      GOTO 600
253   A(1)=.2177
      A(4)=.2177
      A(5)=.866
      A(6)=1.6407
      A(7)=2.1520
      A(8)=1.5369
      A(9)=1
      GOTO 600
254   A(1)=.3627
      A(4)=.3627
      A(5)=1.3347
      A(6)=2.4383
      A(7)=2.8469
      A(8)=2.0480
      A(9)=1
      GOTO 600
260   GOTO (261,262,263,264),NRIP
261   A(1)=.031305
      A(4)=.044219
      A(5)=.16335
      A(6)=.698804
      A(7)=.69044
      A(8)=1.66249
      A(9)=.57068
      A(10)=1
      GOTO 600
262   A(1)=.053455
      A(4)=.059978
      A(5)=.27353
      A(6)=.85633
      A(7)=1.12151
      A(8)=1.84353
      A(9)=.90698
      A(10)=1
      GOTO 600
263   A(1)=.12015
      A(4)=.12154
      A(5)=.57829
      A(6)=1.43530
      A(7)=2.12876
      A(8)=2.48288
      A(9)=1.56658
      A(10)=1
      GOTO 600
264   A(1)=.21859
      A(4)=.21884
```

```
      A(5)=.99163
      A(6)=2.25965
      A(7)=3.25896
      A(8)=3.31983
      A(9)=2.13412
      A(10)=1
      GOTO 600
270   GOTO (271,272,273,274),NRIP
271   A(1)=.015660
      A(4)=.01566
      A(5)=.14614
      A(6)=.29999
      A(7)=1.05175
      A(8)=.83139
      A(9)=1.91147
      A(10)=.5684
      A(11)=1
      GOTO 600
272   A(1)=.027253
      A(4)=.027253
      A(5)=.19291
      A(6)=.50381
      A(7)=1.26811
      A(8)=1.35757
      A(9)=2.10317
      A(10)=.90750
      A(11)=1
      GOTO 600
273   A(1)=.064585
      A(4)=.064585
      A(5)=.37852
      A(6)=1.06716
      A(7)=2.07911
      A(8)=2.60152
      A(9)=2.79095
      A(10)=1.58543
      A(11)=1
      GOTO 600
274   A(1)=.12595
      A(4)=.12595
      A(5)=.68572
      A(6)=1.85737
      A(7)=3.29509
      A(8)=4.04874
      A(9)=3.73515
      A(10)=2.19127
      A(11)=1
      GOTO 600
280   GOTO (281,282,283,284),NRIP
281   A(1)=.0078288
      A(4)=.011058
      A(5)=.056474
      A(6)=.32070
      A(7)=.47185
      A(8)=1.46650
      A(9)=.97189
      A(10)=2.16057
      A(11)=.56696
      A(12)=1
      GOTO 600
```

```
282   A(1)=.013831
      A(4)=.015519
      A(5)=.097971
      A(6)=.41410
      A(7)=.79334
      A(8)=1.74349
      A(9)=1.59161
      A(10)=2.36061
      A(11)=.90788
      A(12)=1
      GOTO 600
283   A(1)=.034141
      A(4)=.034536
      A(5)=.22902
      A(6)=.78720
      A(7)=1.67634
      A(8)=2.79177
      A(9)=3.06245
      A(10)=3.08426
      A(11)=1.59803
      A(12)=1
      GOTO 600
284   A(1)=.070308
      A(4)=.070390
      A(5)=.44639
      A(6)=1.42188
      A(7)=2.93459
      A(8)=4.41480
      A(9)=4.80689
      A(10)=4.10960
      A(11)=2.23073
      A(12)=1
      GOTO 600
300   GOTO (300,320,330,340,350,360,370,380),NPOLE
320   A(1)=1.61804
      A(4)=1.61804
      A(5)=2.20321
      A(6)=1
      GOTO 600
330   A(1)=2.7718
      A(4)=2.7718
      A(5)=4.86637
      A(6)=3.41750
      A(7)=1
      GOTO 600
340   A(1)=5.25828
      A(4)=5.25828
      A(5)=11.11552
      A(6)=10.07023
      A(7)=4.73057
      A(8)=1
      GOTO 600
350   A(1)=11.21331
      A(4)=11.2131
      A(5)=27.21909
      A(6)=29.36504
      A(7)=17.82010
      A(8)=6.17948
      A(9)=1
      GOTO 600
```

```
360   A(1)=26.6313
      A(4)=26.6313
      A(5)=71.9941
      A(6)=88.4667
      A(7)=63.7755
      A(8)=28.7348
      A(9)=7.7681
      A(10)=1
      GOTO 600
370   A(1)=69.2265
      A(4)=69.2265
      A(5)=204.3353
      A(6)=278.3697
      A(7)=228.2392
      A(8)=122.4894
      A(9)=43.3861
      A(10)=9.48609
      A(11)=1
      GOTO 600
380   A(1)=194.054
      A(4)=194.054
      A(5)=617.007
      A(6)=915.511
      A(7)=831.692
      A(8)=508.541
      A(9)=215.592
      A(10)=62.3170
      A(11)=11.3223
      A(12)=1
      GOTO 600
390   A(1)=1
      A(4)=1
600   RETURN
      END
C***************************************************************

      SUBROUTINE SHIFTF(SIGNAL,DELTAT,DELTAF,NSTART,NPOINT)

      REAL          X,PI,DELTAT,DELTAF
      COMPLEX       SIGNAL(0:1024)
      INTEGER       NPOINT,NSTART

      PI=3.1415926
      DELTAW=2*PI*DELTAF

      DO I = NSTART,NPOINT-1,1
            X = DELTAW*I*DELTAT
            SIGNAL(I)=SIGNAL(I)*CMPLX(COS(X),SIN(X))
      END DO

      RETURN

      END

C*****************************************************************

      SUBROUTINE FILTER(F3,TYPE,C,NP,SIGNAL,SCANW,PLENGTH)

      REAL        FN,F3,C(12),SCANW,PLENGTH
      COMPLEX     SIGNAL(0:1024),FILT
```

```
      INTEGER   TYPE,NP

      SIGNAL(0)=SIGNAL(0)*FILT(0,F3,TYPE,C,SCANW)
      SIGNAL(NP/2)=SIGNAL(NP/2)*FILT(NP/(2*PLENGTH),F3,TYPE,
     +                               C,SCANW)

      DO N = 1,NP/2-1,1
            FN=N/PLENGTH
            SIGNAL(N)=SIGNAL(N)*FILT(FN,F3,TYPE,C,SCANW)
            SIGNAL(NP-N)=SIGNAL(NP-N)*FILT(-FN,F3,TYPE,C,SCANW)
      END DO

      RETURN

      END
C********************************************************************

      COMPLEX FUNCTION FILT(F,F3,TYPE,A,SCANW)


      REAL           A(12),F,F3
      INTEGER        TYPE
      COMPLEX        JW

      COMMON /PRE/TO,THETAO

      PI = 3.1415926

      JW = CMPLX(0.0,F/F3)

      IF (TYPE.EQ.0) THEN
                  W=2*PI*F
                 W3=2*PI*F3
                 FILT=EXP(-.346574*(W/W3)**2)*
     +                 CMPLX(COS(W**2/(-2*SCANW)+TO*W+THETAO),
     +                       SIN(W**2/(-2*SCANW)+TO*W+THETAO))
      ELSE
           IF (TYPE.EQ.1) THEN
               FILT=(A(1)+A(2)*JW + A(3)*JW**2)/
     +              (A(4)+A(5)*JW+A(6)*JW**2+A(7)*JW**3+
     +               A(8)*JW**4+A(9)*JW**5+A(10)*JW**6+
     +               A(11)*JW**7+A(12)*JW**8)
           ELSE
                IF (TYPE.EQ.2) THEN
                    FILT=(1/.54)*SIN(.2264*PI*F/F3)/(.2264*PI*F/F3)
     +                     *(.54*SIN(.566*PI*F/F3)/(.566*PI*F/F3)
     +                     +.23*SIN(PI+.566*PI*F/F3)/(PI+.566*PI*F/F3)
     +                     +.23*SIN(.566*PI*F/F3-PI)/(.566*PI*F/F3-PI))
                ELSE
                     IF (TYPE.EQ.3) THEN
                       FILT=(SIN(PI*F*F3)/(PI*F*F3))*CMPLX(
     +                           COS(PI*F*F3),-1*SIN(PI*F*F3))
                     ENDIF
                ENDIF
           ENDIF
      ENDIF

      RETURN
```

```
      ENO
C*****************************************************************
      SUBROUTINE SQUARE(SIGNAL,NPOINT)

      COMPLEX        SIGNAL(0:1024)

      DO I = 0,NPOINT-1,1

             SIGNAL(I)=SIGNAL(I)*CONJG(SIGNAL(I))

      ENO DO

      RETURN

      END
C*****************************************************************
      SUBROUTINE OUMPYOUT(SIGNAL,NPOINT,OELTAT,YMAX,TMAX,Y)

      COMPLEX        SIGNAL(0:1024)
      REAL           Y(0:1024)
      CHARACTER*80   NAMEY

      DO N = 0,NPOINT-1,1
             Y(N)=REAL(SIGNAL(N))
      ENO DO

      YMAX=0

      DO I = 0,NPOINT-1,1
             IF (Y(I).GT.YMAX) THEN
                TMAX=I*OELTAT
                YMAX=MAX(YMAX,Y(I))
             ENOIF
      ENODO

      CALL SGOPEN(9,'WRITE','Y OUTPUT FILE ?',NAMEY,'REAL',NPOINT)
      CALL SGTRAN(9,'WRITE','REAL',Y,NPOINT)

      RETURN

      ENO
C*****************************************************************
      SUBROUTINE TIMEIND(NINDEP,NPOSTF3,NPREF3,TMAX,NI,FILTNUM,
     +                   DELTAT)

      REAL           NPOSTF3,NPREF3

      PI=3.1415926

      TINDEP = 1./(2*NPREF3)
```

```
      IF (FILTNUM.EQ.5) THEN
         TINDEP=NPOSTF3
      END IF

      NINDEP=INT(TINDEP/DELTAT)


      IF (NINDEP.EQ.0) THEN
         NINDEP = 1
      END IF


      NMAX=INT(TMAX/DELTAT)
      N1=NMAX
      TYPE *,'THE NUMBER OF INDEP. TIME INCREMENTS IS:',NINDEP
      TYPE *,'THE TIME INCREMENT OF THE MAXIMUM OUTPUT IS:',N1


      DO WHILE((N1-NINDEP).GE.0)
              N1=N1-NINDEP
      END DO

      NI=N1
      TYPE *,'THE STARTING POINT IN TIME IS:',NI

      RETURN

      END
C*********************************************************************

      SUBROUTINE PMATRIX(KPOINT,P,FK,PREF3,TYPEPRE,POSTF3,TYPEPOST,A,
     +                   B,SCANW,T)

      INTEGER       KPOINT,TYPEPRE,TYPEPOST
      REAL          PREF3,POSTF3,A(12),B(12),SCANW,T,FK(-31:31)
      COMPLEX       P(-31:31,-31:31),FILT,X1,X2,X3,X4

      PI=3.1415926

      DO L = -(KPOINT-1)/2,(KPOINT-1)/2

           DO K = L,(KPOINT-1)/2

                X = 2*PI*(FK(K)-FK(L))*T

                X1 = FILT(FK(K),PREF3,TYPEPRE,A,SCANW)
                X2 = CONJG(FILT(FK(L),PREF3,TYPEPRE,A,SCANW))
                X3 = FILT(FK(K)-FK(L),POSTF3,TYPEPOST,B,SCANW)
                X4 = CMPLX( COS(X), SIN(X) )
                P(L,K) = X1*X2*X3*X4
                P(K,L) = CONJG(P(L,K))
           END DO

      END DO

      RETURN
      END

C*********************************************************************
```

```
      SUBROUTINE DIAGO(KPOINT,D,RK)

      INTEGER         KPOINT
      REAL            RK(-31:31),D(-31:31)

      DO I = -(KPOINT-1)/2,(KPOINT-1)/2
          D(I) = SQRT(RK(I))
      END DO

      RETURN
      END

C***********************************************************************

      SUBROUTINE DPDX(KPOINT,DPD,D,P)

      INTEGER               KPOINT
      DOUBLE COMPLEX        DPD(-31:31,-31:31)
      REAL                  D(-31:31)
      COMPLEX               P(-31:31,-31:31)

      DO L = -(KPOINT-1)/2,(KPOINT-1)/2
           DO K = -(KPOINT-1)/2,(KPOINT-1)/2
                DPD(L,K)=D(L)*DCMPLX(P(L,K))*D(K)
           END DO
      END DO

      RETURN
      END

C***********************************************************************

      SUBROUTINE EIGEN(KPOINT,DPDR,DPDI,DPD,DEGVALS,EGVECT,EGVALS,
     +                 EGVECR,EGVECI)

      INTEGER             KPOINT
      DOUBLE PRECISION    DPDR(-31:31,-31:31),DPDI(-31:31,-31:31),
     +                    EGVECR(-31:31,-31:31),
     +                    EGVECI(-31:31,-31:31)
      DOUBLE PRECISION    DEGVALS(-31:31)
      DOUBLE COMPLEX      DPD(-31:31,-31:31)
      REAL                EGVALS(-31:31)
      COMPLEX             EGVECT(-31:31,-31:31)

      DO L = -(KPOINT-1)/2,(KPOINT-1)/2
           DO K = -(KPOINT-1)/2,(KPOINT-1)/2
                DPDR(K,L) = DREAL(DPD(K,L))
                DPDI(K,L) = DIMAG(DPD(K,L))
           END DO
      END DO

      CALL EISPAC(KPOINT,KPOINT,MATRIX('COMPLEX',DPDR,DPDI,'HERMITIAN'
     +           ,'POSITIVE DEFINITE'),VALUES(DEGVALS),
     +           VECTOR(EGVECR,EGVECI))

      DO K = -(KPOINT-1)/2,(KPOINT-1)/2
           IF (DEGVALS(K).GE.0.0) THEN
               EGVALS(K)=REAL(DEGVALS(K))
           ELSE
```

```
            EGVALS(K)=0.0
         END IF
      END DO

      DO L = -(KPOINT-1)/2,(KPOINT-1)/2
         DO K = -(KPOINT-1)/2,(KPOINT-1)/2
            EGVECT(K,L) = CMPLX(EGVECR(K,L),EGVECI(K,L))
         END DO
      END DO

      RETURN
      END
C**********************************************************************

      SUBROUTINE QANDR(KPOINT,NPOINT,D,Q,S,RMAT,R,EGVECT,PREF3,
     +                TYPEPRE,A,POSTF3,TYPEPOST,B,T,FK,SCANW,PLENGTH)

      INTEGER        NPOINT,KPOINT,TYPEPOST,TYPEPRE
      REAL           D(-31:31),PREF3,FK(-31:31),PI,A(12),B(12),SCANW,
     +               POSTF3,PLENGTH
      COMPLEX        Q(-31:31,-512:512),S(-512:512),R(-31:31),
     +               TEMP(-31:31),RMAT(-31:31),EGVECT(-31:31,-31:31),
     +               FILT

      PI=3.1415926

      DO N = -(NPOINT/2-1),(NPOINT/2-1)
         FN = N/PLENGTH
         DO K = -(KPOINT-1)/2,(KPOINT-1)/2

            X = 2*PI*(FN-FK(K))*T

            Q(K,N) = FILT(FN,PREF3,TYPEPRE,A,SCANW)*
     +               CONJG(FILT(FK(K),PREF3,TYPEPRE,A,SCANW))*
     +               FILT(FN-FK(K),POSTF3,TYPEPOST,B,SCANW)*
     +               CMPLX( COS(X), SIN(X) )
         END DO
      END DO

      DO K = -(KPOINT-1)/2,(KPOINT-1)/2
         TEMP(K)=0
         DO N = -(NPOINT/2-1),(NPOINT/2-1)
            TEMP(K)=TEMP(K) + D(K)*Q(K,N)*S(N)
         END DO
      END DO

      DO K = - (KPOINT-1)/2,(KPOINT-1)/2
         RMAT(K)=0.
         DO L = -(KPOINT-1)/2,(KPOINT-1)/2
            RMAT(K) = CONJG(EGVECT(L,K))*TEMP(L) + RMAT(K)
         END DO
      END DO

      RETURN
      END
C**********************************************************************
```

```
      SUBROUTINE STATS(YMEANN,YVARN,YMEANS,YVARS,KPOINT,EGVALS,
     +                 RMAT,YS)

      REAL              YMEANN,YVARN,YMEANS,YVARS,EGVALS(-31:31),YS
      COMPLEX           RMAT(-31:31)
      INTEGER           KPOINT

      YMEANN=O.
      YMEANS=O.

      DO K = -(KPOINT-1)/2,(KPOINT-1)/2
           YMEANN = YMEANN + EGVALS(K)
      END DO
      YMEANS = YMEANN + YS

      YVARN=0.
      YVARS=0.

      DO K = -(KPOINT-1)/2,(KPOINT-1)/2
           YVARS = YVARS + 2*( RMAT(K)*CONJG(RMAT(K)) ) + EGVALS(K)**2
           YVARN = YVARN + 2.*EGVALS(K)**2
      END DO
      YVARN = YVARN - ( YMEANN )**2
      TYPE *,'THE MEAN OF THE NOISE IS ',YMEANN
      TYPE *,'THE NOISE VARIANCE IS ',YVARN
      TYPE *,' '
      TYPE *,'THE SIGNAL MEAN IS ',YMEANS
      TYPE *,'THE SIGNAL VARIANCE IS ',YVARS
      TYPE *,' '
      TYPE *,'  '

      RETURN
      END
C***********************************************************************

      SUBROUTINE NOISEPDF(KPOINT,NPOINT,DEGVALS,KK,PDFNO,DELTAY,EGVALS,
     +                    POFN)

      INTEGER           KPOINT,NPOINT
      DOUBLE PRECISION  DEGVALS(-31:31),KK(-31:31),PDFNO(0:1024),DUM
      CHARACTER*80      NAME3
      REAL              EGVALS(-31:31),DELTAY,SUMK,POFN(0:1024)

      SUMK=0.
      DUM = DEXP(-50.D00)

      DO K = -(KPOINT-1)/2,(KPOINT-1)/2
           IF (DEGVALS(K).LE.0) THEN
              KK(K)=0.0
           ELSE
              KK(K)=1.0
              DO I =-(KPOINT-1)/2,(KPOINT-1)/2
                    IF (I.NE.K .AND. DEGVALS(I).GT.0) THEN
                       KK(K)=KK(K)/(1.- DEGVALS(I)/DEGVALS(K))
                    END IF
              END DO
              SUMK = SUMK + KK(K)
           END IF
      END DO
```

```
      TYPE *,'AREA UNDER NOISE ONLY PDF = ',SUMK

      DO IY = 0,NPOINT-1
            PDFND(IY)=0.
            Y1=IY*DELTAY
            DO K=-(KPOINT-1)/2,(KPOINT-1)/2
               IF (EGVALS(K).GT.0 .AND. KK(K).NE.0) THEN
                  PDFND(IY)=(KK(K)/(DEGVALS(K)))*
     +                        EXP(-Y1/(DEGVALS(K)))
     +                      + PDFND(IY)
               END IF
            END DO
      END DO

      DO IY = 0,NPOINT-1
            PDFN(IY) = REAL(PDFND(IY))
      END DO

      CALL SGOPEN(9,'WRITE','PDF FOR NOISE ?',NAME3,'REAL',NPOINT-1)
      CALL SGTRAN(9,'WRITE','REAL',PDFN,NPOINT-1)

      RETURN
      END

C*************************************************************************

      SUBROUTINE SANDNPDF(PY,YMEANS,YVARS,NAME4,S1,YS,
     +                    KPOINT,EGVALS,RMAT,YMEANN,NPOINT)

      INTEGER          IFLAG,KPOINT
      REAL             YMEANS,PY(0:1024),S1,PI,NU1,SI,X2,X3,YS,
     +                 EGVALS(-31:31),Y1
      COMPLEX          RMAT(-31:31)
      CHARACTER*9      NAME4

      PI = 3.1415926
      YSTEP=( YMEANS + 10*SQRT(YVARS) )/(100)
      CDMP=.001
      SUBI=10

      DO IY=0,100-1
         Y1=IY*YSTEP
         ZERD=PI/ABS(YS+YMEANN)
         SS=ZERD/SUBI
         IFLAG=0
         NU1=0
         SI=0
         DO WHILE(ABS(S1).GE.CDMP .OR. IFLAG.LT.2)
            X2=O
            X3=0
            DO I = 1,SUBI-1
               X2 = X2 + PINT(NU1 + I*SS,KPOINT,EGVALS,RMAT,YS,Y1)
            END DO
            DO I=1,SUBI
               X3 =X3+PINT(NU1+(I-.5)*SS,KPOINT,EGVALS,RMAT,YS,Y1)
            END DO
               T1 = PINT(NU1,KPOINT,EGVALS,RMAT,YS,Y1)
               T2 = PINT(NU1+ZERD,KPOINT,EGVALS,RMAT,YS,Y1)
            S1 = (SS/6)*(T1 + T2 + 2*X2 + 4*X3)
```

```
          IF (ABS(S1).LT.COMP) THEN
             IFLAG=IFLAG + 1
          ELSE
             IFLAG=0
          ENDIF

          NU1 = NU1 + ZERO
           SI = SI  + S1
      END DO
      IF ( SI.LT.0.0) THEN
         SI=0.0
      END IF
      PY(IY)=SI
      END DO
      CALL SGOPEN(4,'WRITE','PDF S+N FILE?',NAME4,'REAL',100)
      CALL SGTRAN(4,'WRITE','REAL',PY,100)

      RETURN
      END

C**************************************************************************

      SUBROUTINE THRESH(VT,FAR,NPOSTF3,NPREF3,KK,DEGVALS,KPOINT,YMEANN)

      DOUBLE PRECISION     KK(-31:31),DEGVALS(-31:31),X,DUM,PROB,FVT,
     &                     FVT1,FVT2,FVT3,PFA,PNA,VT,VT1,VT2,VT3,TOL,
     &                     TEMP1,TEMP2,VTOLD
      INTEGER              PASS,KPOINT,I1,I3
      REAL                 NPOSTF3,NPREF3,FAR,
     +                     EGVALS(-31:31)

      DUM = DEXP(-50.0D0)
C      TYPE *,'FAR IS: ',FAR
C      TYPE *,'NPREF3 IS: ',NPREF3
      DUM = FAR/(2.D0*NPREF3)
C      TYPE *,'DUM IS: ',DUM
      PROB = 1.0D0 - DUM
C      TYPE *,'THE PROB IS: ',PROB
      VT1= 0.0D0
      VT2= 50.0D0
      TOL = ABS(0.5*(VT1-VT2))

      DO WHILE (TOL .GE. 0.000001)
              VT3 = (VT1+VT2)/2.0D0
C              TYPE *,'VT1 IS ',VT1
              FVT1 = FVT(KK,VT1,DEGVALS,KPOINT) - PROB
C              TYPE *,'1'
C              TYPE *,'VT2 IS ',VT2
              FVT2 = FVT(KK,VT2,DEGVALS,KPOINT) - PROB
C              TYPE *,'2'
C              TYPE *,'VT3 IS ',VT3
              FVT3 = FVT(KK,VT3,DEGVALS,KPOINT) - PROB
C              TYPE *,'3'

C       TYPE *,' '
C       TYPE *,VT1,VT3,VT2
C       TYPE *,FVT1,FVT3,FVT2
C       TYPE *,' '

       IF ((FVT1.GT.0.0.AND.FVT3.GT.0.0).OR.(FVT1.LT.0.0.AND.
```

```
     &       FVT3.LT.0.0)) THEN
              VT1 = VT3
        ELSE
              VT2 = VT3
        END IF

        TOL = ABS(0.5*(VT1-VT2))


      END DO

      VT = VT3
      TYPE *,'THRESHOLD BY BISECTION IS: ',VT
C************************************************************
C    LETS TRY NEWTON'S METHOD
C************************************************************

      TOL=1.0D0
      PFA=FAR/(2.D0*NPREF3)
C      TYPE *,'PFA IS ',PFA
      PNA=1.0D0-PFA
C      TYPE *,'PNA IS ',PNA
      VTOLD =3.D0*VMEANN
C      TYPE *,'THRESHOLD IS ',VTOLD
      DO WHILE (TOL.GT.1.E-6)
         TEMP1=0.D0
         TEMP2=0.D0
         DUM = EXP(-50.D0)
         DO K = -(KPOINT-1)/2,(KPOINT-1)/2
         IF (KK(K).NE.0.0D0 .AND. DEGVALS(K).GT.0.0D0) THEN
         DUM = DUM + 1.D0
         TEMP1 = TEMP1 + KK(K)*(1-EXP(-VTOLD/
     +                 (DEGVALS(K))))
         TEMP2 = TEMP2 + (KK(K)/(DEGVALS(K)))*
     +           EXP(-VTOLD/(DEGVALS(K)))
          END IF
          END DO

         VT = VTOLD + (PNA-TEMP1)/TEMP2
C           TYPE *,'VT IS ******** ',VT
      IF (VT.GT.50.0D0) THEN
         VT = 50.0D0
      END IF

      TOL=ABS(VT - VTOLD)
      VTOLD=VT
      END DO


      TYPE *,' '
      TYPE *,' '
      TYPE *,'THRESHOLD VOLTAGE BY NEWTON-RAPHSON IS: ',VT
      TYPE *,' '
      TYPE *,' '

      RETURN
      END

C*****************************************************************
```

```
      SUBROUTINE   PDETECT(S1,PDET,YMEANN,YVARN,KPOINT,EGVALS,
     +                     RMAT,YS,VT)

      DOUBLE PRECISION     VT
      REAL                 S1,PDET,YMEANN,YVARN,EGVALS(-31:31),YS,
     +                     T1,T2,NU1,NUINT(0:100)
      INTEGER              KPOINT
      COMPLEX              RMAT(-31:31)
      CHARACTER*8          NAME13

      DO I = 0,100
             NU1 = I/4
             NUINT(I) = P1INT(NU1,KPOINT,EGVALS,RMAT,YS,VT)
      END DO

      CALL SGOPEN(13,'WRITE','PDET INT FILE? ',NAME13,'REAL',101)
      CALL SGTRAN(13,'WRITE','REAL',NUINT,101)

      PI=3.1415926
      COMP=1.E-7
      SUBI=10
      ZERO=PI/ABS(YS+YMEANN)
      SS=ZERO/SUBI
      IFLAG=0
      NU1=0.
      SI=0.

      DO WHILE(ABS(S1).GE.COMP .OR. IFLAG.LT.2)
         X2 = 0
         X3 = 0
         DO I=1,SUBI-1
            X2 = X2 + P1INT(NU1+I*SS,KPOINT,EGVALS,RMAT,YS,VT)
         END DO
         DO I=1,SUBI
            X3 = X3 + P1INT(NU1 + (I-.5)*SS,KPOINT,EGVALS,RMAT,YS,VT)
         END DO

         T1 = P1INT(NU1,KPOINT,EGVALS,RMAT,YS,VT)
         T2 = P1INT(NU1+ZERO,KPOINT,EGVALS,RMAT,YS,VT)

         S1 = (SS/6)*( T1 + T2 + 2*X2 + 4*X3)
         IF (ABS(S1).LT.COMP) THEN
            IFLAG=IFLAG+1
         ELSE
            IFLAG=0
         END IF

         NU1 = NU1 + ZERO
         SI = S1 + SI
      END DO

      PDET=.5-SI/PI
      TYPE *,'PDET IS',PDET

      RETURN
      END
C***********************************************************************
```

```
      REAL FUNCTION PINT(NU,KPOINT,EGVALS,RMAT,YS,Y1)

      INTEGER              KPOINT
      REAL                 NU,TEMP1,TEMP2,TEMP3,TEMPA,YS,Y1,PI,
     +                     EGVALS(-31:31)
      COMPLEX              RMAT(-31:31)


      PI = 3.1415926
      TEMP1=0
      TEMP2=1
      TEMP3=0

      DO K = -(KPOINT-1)/2,(KPOINT-1)/2
           TEMPA = 1 + EGVALS(K)**2 * NU**2
           TEMP1 = (ABS(RMAT(K))**2 * NU**2)/TEMPA + TEMP1
           TEMP2 = SQRT(TEMPA)*TEMP2
           TEMP3 = ABS(RMAT(K))**2 *EGVALS(K)* NU**3/TEMPA -
     +             ATAN(EGVALS(K)*NU) + TEMP3
      END DO

      PINT = (EXP(-TEMP1)/TEMP2) * COS((YS-Y1)*NU - TEMP3)/PI

      RETURN
      END

C*********************************************************************

      REAL FUNCTION P1INT(NU,KPOINT,EGVALS,RMAT,YS,VT)

      INTEGER              KPOINT
      DOUBLE PRECISION     VT
      REAL                 NU,TEMP1,TEMP2,TEMP3,TEMPA,YS,PI,
     +                     EGVALS(-31:31),SUM
      COMPLEX              RMAT(-31:31)


      IF (NU.NE.0.) THEN
         TEMP1 = 0
         TEMP2 = 1
         TEMP3 = 0

         DO K = -(KPOINT-1)/2,(KPOINT-1)/2
              TEMPA = 1 + EGVALS(K)**2 * NU**2
              TEMP1 = (ABS(RMAT(K))**2 * NU**2)/TEMPA + TEMP1
              TEMP2 = SQRT(TEMPA)*TEMP2
              TEMP3 = ABS(RMAT(K))**2 * EGVALS(K) * NU**3/TEMPA -
     +                ATAN(EGVALS(K)*NU) + TEMP3
         END DO

         P1INT = (EXP(-TEMP1)/(TEMP2*NU))*SIN((VT-YS)*NU+TEMP3)
      ELSE
         SUM = 0.
         DO K=-(KPOINT-1)/2,(KPOINT-1)/2
            SUM = SUM + EGVALS(K)
         END DO

         P1INT = VT - YS - SUM
      END IF
```

```
      RETURN
      END
C*********************************************************************
      REAL*8  FUNCTION FVT(KK,VT,DEGVALS,KPOINT)

      DOUBLE PRECISION        KK(-31:31),DEGVALS(-31:31),DUM,VT,TEMP
      INTEGER                 KPOINT
      REAL                    EGVALS(-31:31)

      TEMP = 0.0D0
      DUM = DEXP(-50.0D0)

      DO K = -(KPOINT-1)/2,(KPOINT-1)/2

             IF (KK(K).NE.0.D0 .AND. DEGVALS(K).GT.0.0D0) THEN
             TEMP = KK(K)*( 1 - EXP(-VT/(DEGVALS(K)
     +                          ))) + TEMP
             END IF
             DUM = DUM + 1.D0 + TEMP
      END DO

      FVT = TEMP

      RETURN
      END
$
```

REFERENCES

[1] M. Schwarz, W. R. Bennet, and S. Stein, Communication Systems and Techniques, New York; McGraw-Hill, 1966.

[2] B. K. Harms, "An Analysis and Comparison of the Channelized, Acouatoopic, and Frequency Compressive Intercept Receivers," A Ph.D. Dissertation, Kansas State University, Department of Electrical and Computer Engineering, 1985.

[3] J. D. Adam, and J. H. Collins, "Microwave Magnetostatic Delay Devices Based on Epitaxial Yttrium Iron Garnet," Proceedings of the IEEE, Vol. 46, No. 5, May 1976, 794-800.

[4] M. R. Daneil, J. D. Adam, T. W. O'Keef, "A Linearly Dispersive Magnetostatic Delay Line at X-Band," IEEE Transactions on Magnetics, Vol. Mag-15, No. 6, November 1979, 1735-1737.

[5] J. M. Owens, R. L. Carter, and C. V. Smith, Jr., "Magnetostatic Waves, Microwave Saw," 1980 Ultrasonics Symposium, 506-509.

[6] H. Hewitt, "Microacan Revisited--Candidate for the Agile Environment," EW: Sept./Oct. 1976, 49-53.

[7] "Compressive Receivers Making ELINT Comeback," MSN-VECTOR, March 1972, 19-21.

[8] G. L. Movle, R. A. Bale, and T. I. Browning, "A 1 GHz SAW Compressive Filter," 1980 Ultrasonics Symposium, 216-219.

[9] W. D. Daniels, M. Churchamn, R. Kyle, and W. Skudera, "Compressive Receiver Technology," Microwave Journal, April 1986 (175-184).

[10] B. K. Harms, D. R. Hummels, "Calculation of Detection Probabilities for Frequency Compressive Receiver," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-21, No. 1, January 1985, 106-116.

[11] A. J. DeVries, R. L. Miller, and T. J. Wojcik, "Reflection of a Surface Wave from Three Types of ID Transducers," 1972 Ultrasonics Symposium Proceedings, 353-381.

[12] P. C. Meyer, and D. Gunea, "Design and Fabrication of SAW Multipole Filters," 1983 Ultrasonic Symposium, 66-71.

[13] B. Lewis and R. G. Arnold, "Electrode Reflections, Directionality, and Passband Ripple in Wideband SAW Chirp Filters," IEEE Transactions on Sonics and Ultrasonics, Vol. SO-32, No. 3, May 1985, 409-418.

[14] D. R. Hummela, C. Adams, and B. K. Harms, "Filter Selection for Receivers Using Square-Law Detection," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-19, No. 6, November 1981, 871-883.

[15] D. Nichols, "Transform Receivers for ECM Applications," Microwave Journal, October 1982, 113-127.

[16] J. B. Harrington and R. B. Nelson, "Compressive Intercept Receiver Uses SAW Devices," Microwave Journal.

[17] H. R. Stocker, R. Veith, E. Willibald, and G. Riha, "Surface Wave Pulse Compressive Filters with Long Chirp Time," 1981 Ultrasonics Symposium, 78-82.

[18] M. A. Jack, P. M. Grant, and J. H. Collins, "The Theory, Design, and Applications of Surface Acoustic Wave Fourier-Transform Processors," Proceedings of the IEEE, Vol. 68, No. 4, April 1980, 450-468.

[19] W. L. Bangianni, Microwave Journal 17, number 1, 49, 1974.

[20] Stein and Jones, Modern Communication Principles, McGraw-Hill, 1967.

AN ANALYSIS OF A FREQUENCY COMPRESSIVE RECEIVER

by

Thomas Richard Walsh

B.S., Kansas State University, 1981
M.S., Kansas State University, 1983

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1986

ABSTRACT

This thesis investigates the probability of detecting an RF pulse incident on a frequency compressive receiver (FCR). The FCR is considered with and without a lowpass post filter.

A mathematical series representation is used to characterize the signal and the noise incident on the receiver. Then, a matrix representation of the filter output is developed. The probability density function of the output is obtained by taking the Fourier transform of the characteristic function of the output. Given this probability density function, the probability of detection is numerically evaluated as a function of a specific threshold.

The probability of detection is examined for three configurations of the parameters associated with the FCR. Optimization of the FCR in terms of the filter's dispersion time and RF bandwidth is also considered.